A
SYSTEMS ANALYSTS
& DESIGN
READER

By
Lim, Marylyn Grace C.

TABLE OF CONTENTS

DEDICATION

This Systems Analyst & Design Reader is dedicated to Mr. Paul Pajo. This is also dedicated to my family and friends especially to my best friend who is always by my side when I’m doing all of these book reviews, case studies, use cases and SAD paper.

PREFACE

I learned so many things while doing these book reviews, case studies, use cases and SAD paper. Some of them are determination, perseverance and patience. I learned some things regarding startup, that nothings goes on plan, that hobby can also be a startup and that everything is trial and error. I also learned how to analyze systems and how to make a use case narrative out of the use cases. For me, not all of these things can be learn from school or from your professors; some of these are learned from experience. Just like the use cases, you can't make a use case narrative if you didn't go through the process. Almost everything is learned from your own experience.

BOOK REVIEW

Book: Systems Analysis and Design: an Active Approach

Author: George M. Marakas

Reference No.: QA 76.9 S88 M37 2001

BOOK REVIEW # 1

Chapter 1: The Systems Development Environment

Quote: "Anything that can become digital will become digital, and this means that all organizations will ultimately become irreversibly dependent on their information and automation systems and the people who design, develop and maintain them.

Review:

This book discusses about systems analysis and design. Chapter 1 discusses about the systems development environment. It defines what does SAD stands for and what it means. SAD stands for system analysis and design. SAD is a systematic approach to identify problem opportunities, Analyze the information in the organization and design a computerize information system to solve the problem. A system analyst acts as an outside consultant to business organization, supporting experts with in a business and they also act as change agent. They are the ones who bridges the communication gap bet those who need the computer and those who understand technology. This chapter also discusses the four basic skills that a system analyst must have. First is the technical skill. This skill helps the analyst to communicate with wide variety of technical experts. Second is analytical skill. We know that a system analyst is a problem solver. Analytical skills help them in identifying problems and finding solution for it. Third is managerial skill. A system analyst is also a team player who has developed expertise as technical and managerial member of the organization. Not only have they had knowledge on computer but also in management. The last skill that a system analyst should have is interpersonal skill. A system analyst must be a good communicator who can work effectively with all types of people. They should have this skill because they will need it to communicate information to other participants in the project. This chapter also discusses the five types of information systems and systems development. First is the transaction processing system. This is the most prevalent and oldest of the computer-based IS types. Its primary purpose is to automate the capture and recording of information about the transactions that occur during the course of conducting business. Second is the management information system. This is often characterized as an extension of TPS because it accepts as its input the raw date from a TPS and transforms it into reports that help the managers in managing business. Third is decision support system. This system is built to assist the activities that provide support to the decision process. Fourth is the office automation and workgroup management system. This facilitates the analysis and dissemination of information throughout the organization. Lastly, web-based system is designed to facilitate the conduct of business over the internet. The last topic that is being discussed

in this chapter is the systems development life cycle. SDLC is the step on how to create an information system. It describes the steps in project development which are analysis, design, implementation, quality assurance and testing, deployment and the last step is maintenance.

BOOK REVIEW # 2

Chapter 2: So What Is The Problem?

Quote: “For either you know what you are looking for then there is no plan or you didn’t know and then you cannot expect to find anything.”

Review:

This chapter talks about the world of problems as system and, in the process. It also talks about the two primary activities of the system analyst, which are identifying and solving problems. There are two basic elements in problem identification and problem solving; these are the problem and the symptoms. These can be also defined as cause and effect. There are tools designed to assist the system analyst in identifying problems. First is the Ishikawa diagram or fish bone diagram. This diagram is the cause-and-effect diagram that relates the problem under question to the factors driving it. The second framework is the Pieces framework. With the help of this framework, analyst can allocate additional resources to further investigate the possible cause of the symptoms. This chapter also discusses the two system classification, which are closed system and open system. A closed system have a distinguish characteristics of a generally self-contained design while the open system can adapt to changes in both internal and external conditions. Lastly, this chapter discusses about the systems development principle. The 6 system development principles are get the users involved, systems analysis is problem solving, information systems are capital assets , good ideas can be bad ideas, document now and lastly, use the divide and conquer approach. These principles should be followed regardless of the exact development methodology employed.

BOOK REVIEW # 3

Chapter 3: Identification and Selection of Development Projects

Quote: “When schemes are laid in advance, it is surprising how often the circumstances fit with them.”

Review:

This chapter discusses the importance of a formal evaluation and selection process for organizational application development projects. The systems development projects are proposed for two reasons. First, the perception of a problem that can be solve with a computer-based solution. Second, the identification of an opportunity for improvement that can be facilitated by upgrading a current system or designing a new one. There are four evaluation criteria used in the selection of system development projects. First is the potential organizational benefit. This determines the project’s

potential for providing significant, ongoing and measurable benefits to the organization. Second is the strategic fit, this evaluates the degree to which the proposed project fits the existing organizational strategic focus. Third is the level of resource allocation. This is an assessment of the amount and types of organizational resources that must be committed to the proposed project to reach a successful outcome. Lastly, the value chain analysis is a process that involves the assessment of an organization's overall activities associated with IT use in terms of the value it add to the enterprise. The creeping commitment is a concept that suggests the process of selection and development is an incremental one that requires a reassessment of the value of the project at each step of the way. The primary source of new development projects in some organization is the result of a formal, ongoing process called information system planning or ISP. ISP is an orderly means of assessing the information needs of an organization and defining the IS, databases and techniques that best satisfy those needs. The preliminary project feasibility analysis is being done when organizations adopt the creeping commitment approach to system development. The preliminary project feasibility analysis takes a categorical approach to the determination of project feasibility to ensure that the continued application of corporate resources is both relevant and contributing to the organization's goals. There are five categories of project feasibility assessment. First is the technical feasibility, this determines the relationship between the present technology resources of the organization and the expected technology needs of the proposed project. Second is operational feasibility, this determines the degree to which the proposed development project fits with the existing business environment and objectives with regards to development schedule, delivery date, corporate culture and existing business process. Third is human factors feasibility, this determines the relationship between the present human resource base of the organization and the expected human resource needs of the proposed project. Fourth is legal and political feasibility, this identifies any potential legal ramifications resulting from the construction and implementation of the new system. Last is the economic feasibility, this assesses the cost-benefit relationship of the proposed project and its net value contribution to the organization. After having a preliminary project feasibility analysis, they will do the baseline plan. The baseline plan is a deliverable from a successful preliminary feasibility assessment. There are six sections in this document. First is plan summary, this is a overview of the project. Second is formal problem definition. This contains the scope and objectives of the project. The third is system narrative section; this contains information related to the proposed system's expected configuration. Fifth is the feasibility assessment, this is where the outcome of the categorical feasibility studies is reported. The final section is the managerial issues, this talk about various cross-life cycle issues. After having the baseline plan, the steering committee, which composed of members of senior management, systems managers and analysts and representative of various stakeholders, will study the project and they will give the formal approval of the project.

BOOK REVIEW # 4

Chapter 4: System Requirements Determination

Quote: “Truth comes out of error more readily than out of confusion”

Review:

In developing an analysis strategy, there are two approaches being use in viewing a system. The first one is called object-oriented perspective. In this approach, the system is being viewed as a collection of objects that interest with each other. The second approach is the process-oriented perspective. This approach tells us that a system can be viewed as a collection of process performed by people in an automated fashion by computers.

There are three “I”’s in requirements determination. These are impertinence, impartiality and insight. The first of the three “I” is impertinence, this characteristics says that an analyst should question everything and take nothing for granted. The second one is the impartiality; it says that an analyst should be able to consider all issues raised by the system stakeholders to provide the best solution for the organization. And the last characteristic is insight. The concept of insight suggests three things to remember. 1.) An analyst must make the assumption that anything is possible. 2.) They should pay constant attention to detail during requirements determination. 3.) Analyst should think creatively during requirements determination. During the requirement determination, analyst sometimes does commit mistakes. There are four common mistakes in requirements determination. The first common mistake in determining requirement is assuming that the end users of a system are aware of all of their information requirements. Second is collecting requirements from each end user instead of all end users. The third common mistake made by analyst is that they ask the wrong questions so they get less useful responses. The last common mistake is failing to take advantage of the benefits associated with trial and error.

In requirement determination deliverables, there are six good requirement characteristics that should be followed. First, testable and verifiable, this suggests that requirement must be written such that they represent the acceptance criteria for the new system. Second, justifiable, requirement should be necessary rather than simply desirable. Third, unambiguous, requirement should be stated such that multiple interpretations are excluded. Fourth, modifiable, requirement should allow changes in the business environment. Fifth, consistent, requirement should not be in conflict with any other stated requirement. Lastly, hierarchically traceable, requirement should contain a single system attribute and should be traceable back to a higher level requirement.

There are two categories for requirement determination techniques that have been identified based on their relative recency of adaptation as information-gathering approaches. The two categories are traditional method and modern method. In the traditional method, there are four types of information gathering methods. 1.) Direct Interview is meeting with individuals to ask questions about their roles, responsibilities and needs for the current and proposed systems. 2.) Questionnaires and surveys; the analyst submits written, structured question to selected individuals to gather information. 3.) Direct observation is observing individual, process and events to determine the facts

surrounding a particular process within a business environment. 4.) Archival document analysis is reviewing recorded organizational documents. The second category in information-gathering approaches is the modern method for requirement determination. There are two kinds of approaches use. The first is JAD. JAD's main objective is to bring the stakeholders of a proposed system together and use that gathering to facilitate and expedite the collection and determination and consensus on requirements for a new system. The second approach is the iterative prototyping. This helps analyst to quickly convert the basic requirement of a system into a limited working model that can be viewed and tested by the end users.

BOOK REVIEW # 5

Chapter 5: Modeling the Process & Logic

Quote: "I'm never content until I have constructed a mechanical model of the subject I'm studying. If I succeed in making one, I understand; otherwise I don't."

Review:

As a system analyst, there are several tools used to clear the system requirements by representing them as a set of models. This chapter talks about one of the tools used by system analyst, the Data Flow Diagram (DFD). DFD is a logical process-modeling that is used to represent different processes and data flows associated with both the current and proposed system.

In logical process modeling, models are simplified representations of reality. The physical model represents "how" something happens while the logical model tells "what" a does w/o any constrictions of how that might be accomplished. In system analysis, there are 3 benefits of using a logical modeling approach. First, logical models reduce the risk of missing important business requirements of the system. Second, it makes easier for the system analyst to communicate w/ end users & system stakeholders. Lastly, this reduces the biases associated w/ the way the current system is implemented.

DFD has 4 requirements. These are data flow, data store, process & entity. Data flow represents data that is in motion. While the data store represents data at rest. A process can transform data into another form or creates a new data or assembles the data into a useful output. Lastly, an entity interacts w/ the system but resides outside the system boundary. Using these 4 components, a system analyst can represent a complex system & understand the details not only the objectives of the system but also they will understand the system's capabilities of achieving it.

A system analyst creates a DFD hierarchy so that they can understand the complexity of a system & analyze it from a different perspective. There are 3 levels of DFD hierarchy. First, context level diagram, this diagram identifies the system boundary & its relationship to any source that may interact w/ it. Second, level-0 DFD, this represents the major processes contained in the system, the sequence of processes, the data stores accessed by the processes & the sources that interact w/ the system. Lastly, level-1 to level-n DFD, this diagram is created when the level-0 diagram is already completed & verified to be an accurate representation of the system.

Lastly, this chapter talks about the basic modeling. Logic modeling is used to model the sequential or temporal logic contained in the process. There are 4 common logic modeling tools use. These are structured English, the decision table, decision tree and state-transition diagram. First, the structured English uses action verbs & nouns phrases to describe logic sequence & events. Second, decision table, this is a diagram of all the logic & possible outcomes associated w/ the process. Third, the decision tree, this represents the logic in a manner that is like a tree trunk & its branches lying on its side. Lastly, the state-transition diagram, this is useful for modeling logic that is time-dependent or temporal.

BOOK REVIEW # 6

Chapter 6: Modeling the Data Conceptual & Logical Data Modeling

Quote: "Man is a tool-making animal."

Review:

This chapter talks about data modeling. Data modeling illustrates the nature of the data used by a system in terms of the rules and their relationship by which it operates. In data modeling, there is one most common format being used, and that is ERD. ERD represents the conceptual data requirements for the proposed system that identity the rules and interrelationship among data. In ERD, there are 5 basic symbols used for constructing entity-relationship diagrams. 1.) Entity. This is represented by a rectangle. Entity is any data that the user wants to store. There are 2 important characteristics of entities. A.) Entities can be distinguished from other entities by some set of identifying characteristics or attributes. B.) Entities represent a class of things that can occur more than once with the organization. 2.) Attributes. This is represented by an oval. An attribute contains a characteristics or descriptive property of an entity or a relationship. 3.) Relationship. This is represented with a diamond. Relationship is an association between one or more entities that has interest to the organization. Relationships are identified by 3 basic characteristics of complexity. A.) Cardinality. This indicates the number of times that an entity in a relationship that are associated with each instance of the other entities in that relationship. B.) Optionality. This represents the degree which a relationship is mandatory or optional. C.) Degree. This represents the number of entities participating in a relationship. 4.) Associative entities. This is represented by a rectangle with a diamond in it. Associative entity inherits the primary key from more than one entity. 5.) Multivalued attributes. This is represented with an oval and another smaller oval overlapping it. Multivalued attribute is a special type of attribute; this can take multiple values on a single entity instance.

In this chapter, you should also remember some key terms like entity type, entity instances and key attributes. Entity type is a collection of entities that share a common property or attribute. Entity instance represents a single, unique occurrence of a member of an entity type. Key attributes identifies and differentiates one entity instance from other possible entity instance of an entity.

Logical data modeling is the process wherein the initial identification of the component elements of the conceptual date model is change into a useful logical model. The logical data modeling has 3 purposes. 1.) Formal structuring of the data in a more stable and desirable from through normalization. 2.) Development of a data model in a form that allows the actual data needed by the organization. 3.) The development of a data model than can construct a physical database design.

Lastly, this chapter lets you know if a data model is a good data model. A good data model should consist of 8 characteristics. 1.) pictorial 2.) rigorous & specific 3.) top-down decomposable 4.) provides focus 5.) minimally redundant 6.) transparent 7.) easily navigated 8.) predicts the final system.

BOOK REVIEW # 7

Chapter 7: CASE Tools and Joint and Rapid Application Development

Quote: "Engineers are not superhuman. They make mistakes in their assumptions, calculations and conclusions."

Review:

This chapter discusses 3 things. These are CASE tools, joint application development (JAD) and rapid application development (RAD). CASE tool stands for computer-aided software engineering tool. CASE tool helps a software engineer to maintain and develop software. The main objective of CASE tool is to support each phase of the SDLC with a set of laborsaving tools. These tools are upper case, lower case and life-cycle–spanning case. Upper case focuses on supporting the early phases of the life cycle by providing automated assistance. Lower case focuses on the implementation phases of the life cycle. Lastly, life-cycle-spanning case supports the whole process of analysis, design and implementation. There are 4 important components in modern CASE tools. First, central repository, this represents the central storage and retrieval location for information necessary to analyze, create , design and modify a software application development from the start of the process. Second, modeling and diagramming tools, this assists the analyst in the development and testing of logical models of processes and data structure. Third, prototyping and transformation tools, this enables a quick conversion of requirement defined information a default database and application designs. Fourth, documentation generators, this enables phase-specific documentation from the data contained within the central repository.

Second thing that is to be discussed in this chapter is the joint application development (JAD). JAD helps bring formal structure and increased effectiveness to the requirements-gathering activities of the early phases of the SDLC and at the same time it improves the efficiency of the information-gathering process. Some of the advantage of JAD is that it helps improve the quality of the information-gathering activities while managing the time and resources needed to gather the information in a more efficient manner. Second, this enhances the development of a shared understanding among the

system stakeholders with regards to the system. And lastly, JAD helps improve user motivation and performance.

Lastly, this chapter talks about RAD or rapid application development. RAD is a method that helps organization to develop and deploy a system more quickly and at the time it maintains quality and reduces development costs. There are 6 advantages that an organization can get from RAD. First, time savings in the project phases are realizable. Second, this reduces the project cost and human resource requirements. Third, RAD blends well with development efforts where time is gold. Fourth, system design changes can be affected more quickly than with the old SDLC approach. Fifth, the user perspective is presented in the final system with regard to both functionality and interface. Lastly, it creates a strong sense of ownership among all project stakeholders.

BOOK REVIEW # 8

Chapter 8: Moving from Analysis to Design

Quote: "There are many hows but only one what. Never tell people how to do things. Tell them what to do and they will surprise you with their ingenuity.

Review:

As what the title of chapter 8 says, "Moving from analysis to design". This chapter talks about designing a system. After analyzing the current system, it's now time to design a new system. Before a system analyst could design and implement their proposed solution, they should have a design strategy. Design strategy is a process in pursuing the physical development of the system. This is an approach that is being followed to transform the logical system requirement into a functional, physical IS.

There are 8 selection of design strategy. 1.) Generating alternative design strategies. Even if the proposed system is the exact solution to the problem of the current system, there is still a feasibility assessment that is being conducted which includes determining the viability of pursuing the physical system on a number if dimensions. 2.) Do nothing. A system analyst should know the existence of the problem and fully understand its implement a solution to the problem. So that all the hard work of the analyst will be worth it. 3.) Explore all possible non-automated solution. It is important that analyst should give attention to the potential for solving the problem in a non-automated manner. This says that before an analyst inject the complexities of automation to the solution; they must first look to portions of solution that can be affected without computer. 4.) Buy versus make. This is to let the analyst know how to acquire the physical system components, especially the software applications, which led to one of the 3 basic approaches. A.) buy a COTS solution and customize it to fit in the system. B.) develop a custom software application or C.) Outsource the project to an external developer. 5.) COTS. This provides a solution that is well tested, proven and readily available than a custom-developed application. 6.) Custom software development. If an analyst builds the software from scratch it would be advantage for him because he has the complete control over the look, feel and functionality of the new system. 7.) Outsourcing.

This can help bring opportunities to bring increased value to the business. 8.) Hardware design strategy issues. Other than focusing software development, analyst should also focus on issues related to hardware, as it relates to the application software under development and the business needs of the organization.

The dimensions of system feasibility allow the analyst to insure that the business case for the development and implementation of the new system is made from a fact-based perspective. In order to do this feasibility analyst, there are 5 primary dimensions being looked at. 1.) Technical feasibility. This determines the practicality of a specific technical solution and the availability of technology to implement it. 2.) Operational feasibility. This focuses on issues related to how well the new system will work within the organization and how the users feel about it. 3.) Human factor feasibility. This focuses on issues regarding system usability and end-user training. 4.) Legal and political feasibility. This focuses on issues that may be associated with the new system or any political impacts likely as a result of its deployment. 5.) Economic feasibility. This identifies the financial and net economic impacts to the organization of the new system.

BOOK REVIEW # 9

Chapter 9: Designing the Files and Databases

Quote: "The commercial world is a great reality check for new ideas."

Review:

This chapter focuses on physical file and database design. Database schema is the physical model of the new system. This also represents the technical implementation of the logical model. During this process, there are 2 primary goals that should be accomplished. 1.) Accurately translate the logical relations into a comprehensive technical specification for the files and database. 2.) Specify the storage and technologies needed for the data to be contained within the system.

In designing and specifying fields, there are 2 things to be remembered. 1.) Field type. This is the principal database. 2.) Data type. This is the specification of how each field in the file represents the data contained within it. 3.) Field representation. This determines the characteristics about each field and records it in the data dictionary for the system. 4.) Date integrity. This focuses on processes and procedures that are intended to facilitate an atmosphere of trust. 5.) Key integrity. This insures that every file has a unique primary key. 6.) Domain integrity. This insures that no data element in a file on database takes on a value that is outside its range of legal. 7.) Referential integrity. This insures that the value of the field is limited to the range of existing values of another field in another file.

In designing the physical record, there are 3 things to be remembered. 1.) Determine the arrangement of the fields contained in each record of the file. 2.) Make a decision on the recording length. 3.) Consider the information process, retrieval and display methods.

In designing and specification files, there are 3 things to be remembered. 1.) File access methods. This contains the methods used to access a particular record in a file. 2.) Organization technologies. This focuses on physical arrangement of records within a file in the storage media. 3.) File volumetric. This addresses the question of exactly how much data the system needs to manage.

Lastly, this chapter discusses about designing the database. Databases architecture determines the structure of the database system. There are 4 basic architectural approaches being used. 1.) Network database model. This approach tells us that in each file in the database may be associated with any number of subordinate or superior files. 2.) Hierarchical database model. This links the files within the database in the form of hierarchy. 3.) Relational database model. This represents a table in which each column represents a data field and the row represents a single record in that file. 4.) Object-oriented database model. This summarizes both the attributes associated with an object and the methods that operate on the objects in a structure known as object class.

BOOK REVIEW # 10

Chapter 10: Designing the System Output

Quote: "Computers are useless, they can only give you answer"

Review:

Output is one of the most important things in SAD. This represents the universal goal of IS. Now, in this chapter, this will discuss about the designing of the system output. The primary objective of an output design is to create an output that presents the desired information to the users in an understandable and usable fashion way in requiring the least effort on the part of the users to obtain it.

There are 8 characteristics of a system output. 1.) Purpose. In system output, an effective output design is dependent on knowing how the output is used. 2.) Recipient. Knowing who will be the recipients of the output can serve as information to the design. 3.) Frequency. This is important because if a designed output isn't obtained when it's needed, then it's usefulness in assisting a given decision process becomes questionable. 4.) Distribution. In distribution, output is transferred from one place of data stored to another place of data stored. 5.) Date Source. Before an output can be designed, the data must exist within the system or accessible to the system. 6.) Format. This is the material effect on its usefulness to the user. 7.) Media. In media, output generated must be determined and designed and analyzed. 8.) Controls. Any contol associated with the generation and distribution of the output must be enforced prior to the generation of the output.

There are 2 types of output. First is the external output. This leaves the boundary of the system and serves to confirm a system action and triggers an action on the part of a recipient. Second is the internal output. If in external output, it leaves the system

boundary, in internal output, it stays within the system boundary and it supports the organizational responsibilities and activities of the users of the system.

There are different devices use to generate system output and media available for recording the output. 1.) Hard Copy. This is also called a paper-based output. This is the common output media used. 2.) Screen Output. This has more advantage over hard copy. Like increased accessibility, reduce storage costs, provides real-time editing and modification, avoids unnecessary storage resources and avoids redundant and out of date printouts. 3.) Audio and video Output. This has the potential of richness in the communication of information that no other output media can. 4.) Computer output microfilm. This contains a roll of photographic film that contains a volume of hard-copy in miniature form. 5.) Robotic Output. Here, robots are used as vehicles for computer outputs intended to any application requiring precise and accurate handling of physical components.

BOOK REVIEW # 11

Chapter 11: Designing the Inputs and User Interface

Quote: “We’ve all heard that a million monkeys banging on a million typewriters will eventually reproduce the entire works of Shakespeare. Now, thanks to the Internet, we know this is not true.”

Review:

After designing the system output, it’s now time to design the inputs and user interface. This chapter discusses things related to the capture and input of the data needed to create the required system outputs. Also, this chapter focuses on the design characteristics of well-designed user interfaces. This chapter also has topics of different controls available for the analyst to insure high integrity and accurate input data.

There are 5 basic categories of human-computer interaction. 1.) Command dialogue. This approach lets the user to initiate all dialogue by issuing instruction to the computer via a structured syntax of commands. 2.) Menus. This shows a list of available alteration for the users that are relevant to the task being performed. 3.) Icons and buttons. This shows an iconic menu that helps the user to select and execute commands by pointing and clicking at graphical buttons/icons located on the menu bar. 4.) Input screens and forms. This uses a screen-based form that helps the user to fill in the needed information in the space provided in such a way that all data are retrieved in a logical and easy-to-read manner. 5.) Natural language recognition. This approach uses a voice synthesis technology to help the user to interface with the computing device by using natural language and voice commands.

There are 3 common input devices used. 1.) Biometric input devices. Thru this device, unauthorized access to the system can be prevented. This refers to an automatic identification of a person based on his/her behavioral characteristics. 2.) Optical input devices. This includes devices like barcodes, optical mark reading and optical character

recognition. Barcodes are used so that it will allow data to be collected faster and accurately. Optical mark reading senses marks in predefined areas of a special form. Optical character recognition scans any form of document as long as it contains numbers, letters or marks. 3.) Smart cards. This allows a huge amount of information to be stored, accessed and processed whether it is online or offline.

There are 4 guidelines to be remembering in input and interface design. 1.) human-computer interaction. This includes desktop metaphor which creates an electronic environment that allows direct manipulation of the multiple items on the desktop. This also includes document metaphor which helps the user to easily navigate between documents. Lastly, this includes dialogue metaphor which creates an electronic version of a conversation using natural language. 2.) Data entry issues. This includes consistency that is the most important characteristics of a good data entry. 3.) Providing user feedback. This includes a feedback loop which is a closed path that maintains a prescribed relationship between output signals and input signals. 4.) Dialogue design. This defines the conditions under which input is captured and conveyed.

Lastly, this chapter talks about designing the input controls. There are 2 basic categories in designing input controls. 1.) Data controls. This helps prevent errors in the system. 2.) Access controls. This prevents unauthorized person from accessing the system.

BOOK REVIEW # 12

Chapter 12: Designing the System Internals

Quote: “Things should be made as simple as possible, but not any simpler.”

Review:

This chapter talks about the things associated with internal design of the system. Modular design is an approach that is used in software design. Modular design is an approach that is used in software design. This approach helps analyst to decompose a large and complex software application a large and complex software application into a smaller interrelated components called modules. Modules are a group of executable instruction that has a single point of entry and exit.

There are 4 important principles of good internal design that should be remembered. 1.) System Factoring. This principle said that the system should be decomposed into small modules that conforms both in size and cohesion guidelines of good design. In system factoring, there are 2 approached used. First is bottom-up approach. This identifies the process needed to be a part of the system then moves forward to code each process as a module that interfaces with all other processes. Second is top-down approach. This views the hardest possible sense then decomposed it into subsystem that works together to efficiently and effectively reach the objectives of the system. 2.) Module Span. This principle states that a parent should not exceed 5-7 children, subordinate and modules. 3.) Module cohesion. This is a measure of

completeness. There are instructions contained within a module that pertains only to the function. 4.) Module Coupling. This states that the modules that are dependent to each other should be minimized and that the amount of communication between dependent modules must also be minimized.

There are 7 types of cohesion. First is functional cohesion. In this type, the modules are designed so that the instructions would work collectively to accomplish a task. Second is sequential cohesion. This defines as the relationship between an instruction and the next in a given module. Third is communicational cohesion. This happens where a module is designed so that the instructions accomplish tasks that use the same data but the sequence of the instructions isn't critical to a successful outcome. Fourth is procedural cohesion. This happens when modules that have instructions grouped together because of some common relationship based on time. Sixth is logical cohesion. This happens when instruction are related to each other only because they are in the same logical class of functions. Seventh is coincidental cohesion. This happens when modules that have instructions that have no relationship to each other.

There are 4 types of coupling. First is data coupling. This states that the dependency among 2 modules is limited to the fact that they pass data between them. Second is stamp coupling. This states that data are passed among modules in the form of data structure. Third is common coupling. This states that to consider 2 modules as common coupled, they should both refer to the same global data area. Fourth is control coupling. This tells us that control information is passed among modules.

Hierarchical structure diagram displays the relationships of the modules to each other and displays the flow and processing of data among the modules of the system. This is a top-down approach to system development using modules. The difference of hierarchical structure diagram and data flow diagram is that in DFD the intended audience is composed of business managers and end users while in HSD the intended audience is the application programmers.

BOOK REVIEW # 13

Chapter 13: Implementing and Maintaining the System

Quote: "There are 2 ways of implementing a software design; one way is to make it so simple that there are obviously no deficiencies and the other way is to make it so complicated that there are no obvious deficiencies."

Review:

This chapter is the last topic of systems analyst and design. After designing the system internals, we are now going to do the final step which is the implementing and maintaining the system. First, we should know what system implementation is. System implementation is an activity that focuses on ensuring that the new system is functional and operational.

The process of system implementation is divided into 3 categories of activities. 1.) Application testing and user acceptance. During the implementation process, testing activities are carefully performed to insure that every aspect of the new system has been checked and rechecked. The acceptance testing is use by end users to verify that the delivered and installed product is ready to be put into production use. 2.) User training and final documentation. This is where the final update of the documentation happens. The documentation is intended to provide the users and administrative with the information they need to operate the system. 3.) System installation and conversion. This refers to the activities and processes connected with replacing the existing system with the new system.

What is post implementation activity? Post implementation activity is the single most costly phase of activity in the life-cycle-approach to systems development. Its primary objective is to correct errors or faults in the system, provide changes to effect performance improvement or to adapt the system to changes in the business environment.

After implementing the new system, it's now time to maintain it. There are 4 categories in systems maintenance. 1.) Corrective. This involves fixing bugs and logic errors that were not detected during the implementation testing period. 2.) Adaptive. This involves modifying existing functions or adding new features to accommodate the changes in the business environment. 3.) Perfective. This involves changes made to an existing system that are intended to improve the performance of a function. 4.) Preventive. This involves activities that intended to reduce the chances of a system failure or to extend the capacity of an existing system.

CASE STUDY

CASE STUDY # 1

PayPal

PayPal is an e-commerce business that allows payment and transfers to be made through the internet. But before PayPal became a successful e-commerce business it undergoes so many problems. I admired Max Levchin, cofounder of PayPal; he was able to surpass all his problems.

The first startup idea that Max Levchin thought of was cryptography software and a service for transmitting money to PDAs. After so many change came, they finally decided to have a web-based payment system. But before it became successful, they encounter problems like fraud and scalability. Because of Max Levchin's determination and perseverance, he was able to find a solution to his problem. Now, he is very successful and PayPal is being used everywhere.

From that story, I learned that to be a successful startup founder, you should have the determination to build things, not just for yourself but also for the benefit of other people. Next is that a good founder should be adaptable with his environment. He should know that nothing goes according to plan, there are still will be changes on it. Lastly, it is hard to do a startup all by yourself. Sure you can do things alone but it is hard to accomplish things just by yourself. Having a good cofounder is having someone you can rely on in a fundamental way.

CASE STUDY # 2

Hotmail

Hotmail is a free webmail service of Windows Live provided by Microsoft. Like PayPal, Hotmail also encountered so many problems before it became famous to everyone.

Sabeer Bhatia and Black Smith were the founders of Hotmail. Before, their first startup idea was a web-based personal database called Java Soft. But their main problem is that their employer's firewall prevented them from accessing their email accounts. In order to solve this problem, they thought of making an email account that can be accessed anywhere through a web browser. On the process of making Hotmail, they encountered problems like funding because many of the VCs turned them down because the VCs were thinking that how can they make money if they are going to give it for free. But eventually Sabeer Bhatia and Black Smith were able to solve their problem and now billions of people are now using Hotmail.

I learned that it's hard to find someone to fund your project especially if it is unfamiliar to them. But it doesn't mean that if one rejected you, you will already give up. A good founder should always have a backup plan in order to succeed and if the founder

has the determination and perseverance he can do anything and he can be successful even in a short span of time.

CASE STUDY # 3

Apple Computer

Apple Computer is a very known company worldwide. This was founded by Steve Jobs and Steve Wozniak. The first product of Apple was Apple I. Apple I was the first computer built by Steve Wozniak for Apple. Back in his high school days, he always wanted to have a computer. Steve Wozniak doesn't have money to buy a computer because it cost as much as a house. So in order to have a computer, he built his own computer using chips. Steve Wozniak was able to build his own computer. One good thing about Steve Wozniak is that he doesn't have any money to buy the parts he needs but he could come up with a good end product that no one done it before. He became successful because he tries to make valuable what he's good at and it wasn't the motivation that he was after. All he wanted is to remain the person that he would have been with Apple.

I realized that sometimes when a person became successful they tend to change personalities. But not with Steve Wozniak, after becoming successful, Steve Wozniak still remains the person he was, that's why until now, he is still successful in his life.

CASE STUDY # 4

Excite

Joe Kraus together with his five Stanford classmates started Excite. Their first idea was developing technology for information search and retrieval, and then it became a web search that made their site fourth most popular site on the web in the late 1990's. But before their company becomes popular they encounter problems like financing and dealing. Nobody wants to invest in their project because the investors were thinking that how are they going to make money with it. Why would people use the search engine more than once if they could just bookmark the site and never go back to the search engine again. But it was luck that made them popular. At first Netscape wasn't going to accept the deal with Excite but because MCI couldn't deliver its service on time, that's why Excite got the deal.

In this story, I learned that dedication and hard work plays an important role in startups, without it, I don't think any of the startup founder would be successful in their business. I also think that luck is also a big thing because without it there will be no Excite today. plays an important role in startups, without it, I don't think any of the startup founder would be successful in their business. I also think that luck is also a big thing because without it there will be no Excite today.

CASE STUDY # 5

Software Arts

Dan Bricklin and Bob Frankston founded Software Arts. This company produces VisiCalc which was the first electronic spreadsheet. Before, spreadsheet used to be made on paper. Dan Bricklin thought of the idea of having an electronic spreadsheet when he was watching his professor at Harvard Business School create a financial model on a blackboard. When the professor found an error he had to erase and rewrite a number of sequential entries in the table. So Dan Bricklin thought of how convenient it would be if they could make an electronic spreadsheet that would make their lives easier.

At first only few people would appreciate the idea of having an electronic spreadsheet, because not all people are into business. But later on, more and more people were using it.

One thing I noticed is that some of the company today like Apple, Excite and Software Arts started developing their product in a garage. At first they were only thinking of a way on how to solve their problem or they just want to satisfy their needs and then it became a product that is widely use.

CASE STUDY # 6

Blogger

Evan Williams is an entrepreneurial person and a web geek. He had started a couple of several of companies. He was a dropped out student because he thought that he don't need to have a degree because he won't going to try to get a job with anyone. But then, he works as a web developer on contract for about a year and a half and has worked in various companies like Intel and HP.

While working he started thinking to start another company. He had visions of raising money and building something cool. Then he thought of Pyra, his first product. The idea of Pyra was the personal and project information management system; to build projects for clients around their interests and help them organize their work and personal information. Pyra is a web application where you could put your stuff, things you are thinking about, things you had to do, things you wanted to share with other people. Evan Williams shared his thoughts to his friend, Meg Hourihan. Meg was so excited about Evan's plans and asked him if she could start it up with him. Evan agreed with her. Both of them worked very hard on their product. They were working on it while working for other companies in order to fund their product. During their first year it was entirely self-funded, but eventually they have to quit their jobs in order to work full-time in their product. They hire their first employee, Paul Bausch. Most of the coding was done by Paul Bausch. Meg and Evan had an internal blog which they called Stuff. This is where they collected everything and it became the center of Pyra. At first, it was only for personal use, but later on, they thought of posting an external company blog. They were the first company to have a blog on their site. So they decided to turn the blog into a product. But the problem is that they don't have enough resources for two products because they don't have enough funds and there were only three of them who are making

the product. Then they decided to launch Pyra. It was successful and more and more people are starting to use it. Evan and Paul continued what they had started with Blogger. Paul and Evan improved the Blogger and launch it. It was a success. Many users where using it.

At first, Blogger was completely free of charge and there was no revenue model. But there came a time that the company was dying. Evan had no money to pay his employees. At first, the employees continued to work without pay for weeks or in some cases months; but eventually it could not last, and eventually, the day came were in Evan was the only came in the office, all of his employees including his cofounder, Meg Hourihan left the company. At that night, he also broke up with his girlfriend. He felt so bad about what happened. But the good news was that the Blogger was still running and with no employees. Evan ran the company all by himself until he was able to secure an investment by Trellix after its founder Dan Bricklin became aware of Pyra's situation. After what happened, Pyra became successful little by little.

I admire Evan Williams so much because he was able to face all his problems. He didn't give up the company even if all his of his employees left him and he is the only one running the company. He kept working and he was able to pass through it. Running a company all by yourself without any employees is a very hard thing to do, if I were at his place maybe I would have left the company because I don't think I could handle such pressure. I never thought that he would be able to do it. But he proved that nothing's impossible. He is always optimistic and he always thinks that there is going to be a better tomorrow.

CASE STUDY # 7

Iris Associates & Groove Networks

Ray Ozzie is the founder of not just one company but two companies. His first company was Iris Associates but later it was acquired by Lotus Development in 1994. His second company was Groove Networks that was also acquired by Microsoft in 2005.

Before Ray Ozzie founded Iris Associates, he worked on PLATO Notes. Later, he was recruited by Jonathan Sachs and Mitch Kapor to work for Lotus Development to develop what became Lotus Symphony. After that, Ozzie left Lotus Development and founded Iris Associates. He always wanted to develop his own software but the problem is that he couldn't find funding. Mitch Kapor and Jonathan Sachs decided to invest on the product which Ozzie created that became known as Lotus Notes. The company's primary funding came from a contractual relationship with Lotus Development. Lotus funded development of Iris Associates' product in exchange for a future option to purchase exclusive intellectual property, marketing and sales rights.

After Iris Associates was acquired by Lotus Development, Ozzie started another company which was Groove Networks. The common theme both Iris and Groove was the fact that the ideas were not based on technology but on a need he saw for users for the product.

One thing I learned from Ozzie is about what he said, “You’re on this mission together. Everyone has to feel that and you have to hire people who are willing to believe in something they are trying to accomplish. But what held people together was the belief that you’re really going to change the world. I think that’s the nature of many startups. You believe that what you are doing is going to have a dramatic impact. You might not exactly know how, but you really have a belief that keeps you going and going through many changes and a lot of uncertainty.” All of these startups started from an idea that you wanted or desire. Just like Ozzie, he always wanted to develop collaboration software of his own. And because of his strong will and determination he became successful. That’s why the people surrounding him believes in everything he does. But one thing that I think that can be prove wrong was that you cannot accomplish something completely on your own. Evan Williams somehow prove that wrong, because the time where in all of his employees leave the company and he was all alone, he was still able to keep the company running and eventually find a way to solve his problems. You just have to have a strong will, perseverance and determination in order to succeed.

We should also appreciate other people’s skills because they are also going to help you start your company. You should have trust with your skills especially in what you’re good at and what other people are good in. We should always remember that people do things for the right reason. People do startups not because of money. They do it because they want to achieve something. And people don’t always succeed in startups. It’s a matter of learning from failure and striving hard for the future. Just like what Ozzie said, “It should about the mission. It should be about changing the world. It should be about how you can impact the lives of users, partners and employees themselves. It’s not just about this big payday.” All of us think of money if we are starting a startup but if we only think of money we won’t be successful and the company will not last long. Stating a startup should be touching other people’s lives and trusting other people’s skills.

CASE STUDY # 8

Lotus Development

Mitch Kapor and Jonathan Sachs founded Lotus Development in 1982 after Kapor left his post as head of development at VisiCorp and selling all his rights to VisiPlot and VisiTrend products to VisiCorp. After Kapor left VisiCorp, he and Sachs produced an integrated spreadsheet and graphics program. This was Lotus 1-2-3. Lotus 1-2-3 could handle larger spreadsheets and added integrated charting plotting and database capabilities.

Lotus started when Kapor helped Eric Rosenfeld write a statistics routine that can ran on the Apple II. This was called Tiny Troll. The publisher of VisiCalc wanted to take Tiny Troll and they will rewrite and clean it up so that it would be a companion product to VisiCalc. Kapor became the product manager and he rewrites the Tiny Troll, which eventually become VisiPlot.

Kapor thought of putting the VisiCalc and VisiPlot on a single disk so that it would be less troublesome for the users. But the people who did VisiCalc weren't interested in it at all. The people who did VisiCalc were not interested because they had serious technical backgrounds nad a bunch of computer science training. They know what they were doing and they had the hot product. While Kapor is not as good as they are.

Kapor left the VisiCorp and met Jonathan Sachs, who had implemented spreadsheets previously but there was no market for that. Kapor convinced Sachs to work with him. Kapor had the royalties and they both have some ideas, so they thought that they might be successful. Then they created Lotus 1-2-3.

One thing that surprised him was that their original business plan called for 3 million to 4 million dollars in sales. But during the first year of Lotus' operations, the company achieved revenues of 53 million dollars and had a successful public offering in 1984. The company tripled in revenue the following year to 150 million dollars. Lotus 1-2-3 became the most powerful spreadsheet program available. The sales were huge, turning Lotus into the largest independent software vendor in the world in just one day. They were very surprised how fast the success was.

How do startups become successful? Maybe it's because of their strong will, perseverance determination and wanting to achieve their goals helped them become successful in life. Sometimes the desire for simple things can help you become successful. At first, Kapor only wanted to have an Apple II then he wanted to try to do something that could stand up well. All of the startup came from failure before it become successful.

The advice that Kapor gave was almost the same as what Ozzie said, "The most important thing for me is, I don't want to work with someone who says, just help me make the business be more successful." This means that the person who only wants the business to be successful only thinks of money. In the business world we should learn to work with entrepreneurs who are personally passionate, committed and believe in what they're doing. Startup is not just about business but it is also about helping other people. We should be the person who is going to create something that actually has value for people in a way that can be a game changer.

CASE STUDY # 9

Yahoo

All of us know that Yahoo is one of the most popular search engines other than Google. The thing we didn't know is how was Yahoo started? Yahoo started in 1994. During the time of Jerry Yang and David Filo, Yahoo was just a collection of links to research papers being run by the two of them. During their free time, they would add some links to new types of information and the site become popular later on. By the end of the year, Jerry Yang and David Filo decided to turn Yahoo into a startup. Tim Brady, Jerry Yang's college roommate, was the one who wrote the business plan for Yahoo.

During the time when Tim Brady was getting his MBA in Harvard, Yahoo grew bigger, so Tim Brady decided to fail three of his five classes just to have time for Yahoo. He used the business plan he made for Yahoo as a final paper for the two classes left. Eventually, he was able to pass it and of course he was able to graduate.

This Yahoo thing happened when Jerry and David were taking their PhD theses and all the information that they need were online. So they need to keep tracking all of them. So they have a big list of information that they needed for their theses, then the EE graduates found out what they are doing so they sent Jerry and David emails saying of they could add this and that to their lists. During their free time, they would add things that their interested in, later on, more and more people are asking that if they could add things that the other people are interested in. as time passed, more and more categories are being added to the list. They were able a huge list, at the right time and in the right place. The VC's notice what Jerry and David was doing. Companies like LA Times, AOL, Microsoft and many others want them to join their companies. Then Jerry and David thought that their project can be not just a hobby but also a business. So they decided to turn hobby into a business. They weren't taking the money that the other companies are offering. Then Tim Brady decided to help them in making a business plan.

As time passed, Yahoo becomes popular. Jerry and David decided to improve Yahoo. They put graphics on it at the same time they also put up advertising. During the time when the company was growing, Jerry and David were having a hard time in managing it so they search for someone to be the CEO. After 6 months if searching, they get Tim Koogle to become the CEO, Jerry and David had tough time convincing people to join Yahoo. They hired a lot of friends but then realized that it was a bad idea because people weren't interest about where the Internet was going. They were just looking for something interesting that they can do so they decided to join Yahoo.

Yahoo grows bigger and bigger. During the time when the Prime Minister of Israel, Rabin, got shot. Jerry and David thought of putting news on the front page. They think that Yahoo is not only a search engine but also a public service in some ways. They use the site to find things on the web and use it to communicate news. Eventually people who are using Yahoo are overwhelmed by it.

This story taught us three things. First, hobby can be a business. Just look at what Jerry and David were able to do. At first they were just collecting interesting things in the Internet then it became a big thing later on. Second, they are not saying that going into a business with friends is really a bad idea. But we just have to choose the right person whom we are going with. Lastly, we should not think that one problem can make your company fall. Maybe if you didn't take action in fixing it, it would really make your company fall. Take the problem as a challenge for you to work hard and improve the things you've done.

CASE STUDY # 10

Research in Motion

Research in Motion or also known as RIM was one of the first companies to appreciate the importance of wireless networks. RIM was founded by Mike Lazaridis. One of RIM's projects was a local area network that ran industrial displays. In 1997, RIM becomes the most admired technology company because he was able to see the potential of mobile email and he was able to a build it and it is being use by Blackberry.

During the third year of Mike Lazaridis' college days, he had been doing computer programming contract work to earn some money to pay for college, then on his fourth year in college, before he graduate, he start his own company. He was able to get a contract that made him so busy that's why he needs to take a leave of absence from school.

Mike Lazaridis and Dong Fregin started RIM. They got a $600,000 contract with General Motors, because they were able to fix a problem that not any other companies could do. The early days of RIM weren't that pretty good. They are worrying about paying the rent. After they heard about the grant from the Canadian government, they started applying for it.

Mike Lazaridis thought of wireless date technology when he was at a conference and someone was talking about putting a wireless data system for Coca-cola. After hearing the story about the wireless data system, Mike thought that it would be an interesting project. Maybe he was lucky during that time when he was just in the right place at the right time when he had received a contract at that point because of his interest.

Cantel was talking about a system that they bought called Mobitex. It was a wireless date system, and they needed someone to make it work. I can say that Mike Lazaridis was very lucky at that period of time. Mike's team were able to write the first wireless protocol software, application programming interface (API). This became a turning point to RIM because no one knew about wireless data.

Another turning point of RIM was that Mike Lazaridis was able to write a program for Blackberry. Blackberry became an interactive pager. They were also able to build an advance pager. This advance pager does look like a pager and has a size of a pager; it even operates like a pager, but the only difference is that the advanced pager is a full-blown two-way email terminal.

That was the time when Blackberry became a pager. So people thought that they needed one of those things. People found it very useful because they could send message as well as receive it. Because of RIM, Blackberry became one of the all-time most famous brands worldwide.

At first, I didn't know what is RIM? And what is it all about? But after reading the story, I learned that RIM, specifically Mike Lazaridis was responsible for making Blackberry the famous brands world-wide. I also learned that in making a project, first you should have a vision on what's going to happen in the future and you should also have faith on it. You should have believed that someday your project is going to happen and it's going to have a value or it is going to be useful to other people. Just like what Mike Lazaridis said, "It's a combination of vision and faith."

CASE STUDY # 11

Marimba

Marimba is a software distribution company that was founded by Arthur van Hoff. But before Arthur van Hoff was able to found Marimba, he was part of the Java development team at Sun Microsystems. The other cofounders of Marimba were his fellow developers from the Java team, Sami Shaio, Jonathan Payne and Kim Poles which was the Java's product manager. Marimba received lots of attention and the company had more than 300 employees back then. By 2002 Arthur van Hoff left the company and started another start up which was the Strangeberry.

This startup started when Jonathan Payne left Sun Microsystems. Arthur tried to convince him to come back, Jonathan said that he don't know if he'll be coming back. But he suggests that he will do a startup with Arthur. They decided to make a startup even though they don't have any idea on what project they are going to do. What they just did was that they find place to make it as an office where they will be working.

The first year of the company was self-funded. The first project they try build was a user interface builder but another company was doing the same thing as theirs and that company was acquired by Netscape. After knowing it, they stop their project.

After what happened, they decided that they will focus on software distribution. They came up with the idea for subscription-based software wherein users don't need to buy the software, they could just subscribe to it and they will get updates automatically. So they announced it to the press that they were doing software distribution, but another problem came out. PointCast had some similarities to what Marimba was doing and they were immediately filed under "push". So they have to explain to the people why they weren't a push company. Eventually, they were able to fix the problem.

There was a point wherein the company gets bigger. It grew from 0 to 300 people that it just becomes a management problem. Kim was replaced by John Olsen who has experience running a company. He was able to make decisions that were very hard for Arthur to make.

They got funding of $4 million from Klein Perkins and another funding from Java. Mike wanted to use the money to buy an espresso machine but the lawyer wouldn't allow it. Few weeks later, there are people who had been working for many years started quitting their jobs and Mike had a hard time in hiring people. Mike was so desperate

about buying an espresso machine, so they bought one. After they bought the espresso machine, everybody's spirits were lighten up. And everyday people would crowd around it.

This story tells us three things. First, in management, it is hard to manage hundreds of people and it's difficult to motivate them. Second, business is not just a business-related expense. You should have an environment that you like and that your people are happy and also they would feel that they are being valued. Lastly, in doing a startup, you shouldn't expect too much that in just a first try your business would already succeed. Whether your startup succeeds or fails, you should have fun while doing it. Enjoy the things you are doing, don't just be a business minded person, you'll become successful if that is really the right time for you to succeed.

CASE STUDY # 12

Gmail

Gmail is Google's web-based email system. This web-based email system was created by the 23rd employee of Google who is Paul Buccheit. He was also the one who suggested the famous motto, "Don't be evil." Paul Buccheit thought of this line when he was in a meeting that was about company values. He was thinking something really different from the usual ones. He also thought that it should be something that it's hard to change. So he thought of the line, "Don't be evil.", because at that time, their competitors are exploiting the users. They would trick the users by putting advertisements on the search results and people actually won't realize that the link was an advertisement. Paul Buccheit said that Google is different from their competitors. They wouldn't do such thing to their users. I think that motto somewhat wants to warn their competitors to stop doing it to their users.

Paul Buccheit created Gmail all by himself and he was able to finish the first version of Gmail in just one day. After building the first version of Gmail, Paul Buccheit thought of creating a better email program than the first version of Gmail. He also thought of the idea of having a search button through your emails because during that time he gets over 500 emails a day. And he was having a hard time looking for emails. So he thought that having a search button would make it better and easier. He made it just like Google search. But the only difference was that it was mad for emails. At first, Paul Buccheit do the all by himself, later on Sanjeev Singh started working w/ him then later on, another person named Jing Lim helped them improved it.

When they launched Gmail, many people thought that it was a joke because that day was April 1 and it was April fool's Day so most of the people didn't believe it. They thought that it was just a joke and they also thought that it was impossible to happen. After launching Gmail, Paul Buccheit worked on another project called Adsense. Adsense was a content-targeted ads and he was also able to finish it in just one evening. At first, people thought that it's impossible to happen but Paul Buccheit proved them wrong. After that, more people took over the project and make it into a real product.

I admire Paul Buccheit because he was a great creator. He was able to create things because he just tries out ideas even if some of the things won't work. And sometimes when something interesting caught his attention, he will do some side projects on it. For me, being an ordinary employee doesn't mean that you can't create a product that will be a big bang in the world. I think the difference in startup and starting w/n a company is that in startup, you will think of what product are you going to do but when you are starting w/n a company, you tend to focus only on your company's product and thinking of a way on how to improve it or maybe you could think of creating something that will be useful for your company using your company's resources.

In the world of IT, we should think what other people will think of your product. You can't just produce a different product or add features into your existing product because some people won't like it. Some people are narrow-minded. They are thinking that new things are impossible to happen. But even if some people think that new things are impossible to happen, you should at least try doing something new and interesting because you don't know maybe you'll get lucky and your product will be a big hit. You can learn a lot of things even if things don't work out. You don't have to be a startup founder just to learn something new, even if you're a normal employee you can learn so many things. The only thing that you should do is to be open-minded.

CASE STUDY #13

WebTV

WebTV is a thin client for surfing the web, using a television as a display. The difference between WebTV and on personal computer is that personal computer uses a computer monitor as a display while a WebTV uses a television as a display. WebTV was created for people who can't afford to buy their own personal computer.

Steve Perlman was liberal arts major. He was a hobbyist and this was his engineering background. He loves to read magazines about technologies. That's where he gets his knowledge in software, hardware, networking and material science. Having this knowledge he was able to build a computer when he was 16 and ha was able to designed graphics display on it. After that, he was able to design a software-based modem when he was in college. His professor gave him an F for his project because his professor doesn't believe that Steve Perlman's project would work. Steve Perlman his professor wrong. He builds the software-based modem and emailed his professor using it saying. "This email is being sent to you on the modem that I designed at Columbia."

After college, he started his first startup which was Catapult Entertainment made a modem for Sega and Nintendo games that would modify the execution of the games. After Catapult Entertainment, settle for a while then he started his second startup which was WebTV. They were able to got Sony to manufactures their product. And they got it exclusive for a year. At that time Philips also wanted to manufacture but Steve Perlman already accepted the offer of Sony, so they told Philips to wait after a year before they could manufacture WebTV. They were able to raise more money, but for some reason, they just about to run out of money. During that time, Steve Perlman didn't tell it to his

employees because he didn't want his people to panic. With that reason, they started to talk to some investors and VCs. Steve Perlman was able to find one VC firm, Brentwood Venture Capital. Jeff Brody put in $4.5 million for the WebTV. After the signing of paper, Steve Perlman got a certified letter from Sony saying, "After due consideration, we've decided not to proceed w/ you in deploying this product." So Steve Perlman just accepts the offer of Philips of manufacturing their product.

After a while, they build a consultant, Spencer Tell of Asia Pacific Ventures, who had done a lot of deals with Japanese companies. This consultant had a personal relationship with the CEO of Sony who was Idei-san. Steve Perlman told the consultant to investigate the letter sent by Sony. After the investigation, Steve found out that the reason why Sony didn't continue the deal was because they thought that it won't be a successful product. Spencer told the CTO to reconsider it, so they decided to get on a private jet to the office of Steve to get the demo. After testing it, there was one website that the CTO went to that didn't work. It was a Japanese character so the website didn't work. One of his engineer, integrated Japanese language support into the code, so by the time when demoed it again, the browser could run English and Japanese characteristics. So they were able to get the deal with Sony. Now that they have 2 companies who manufactures they're product. They were able to raise their money up to $35 million.

With this story, I learned that you don't have to be a computer major just to be able to build software and hardware. Just like Steve Perlman, he was liberal arts major but he was able to build a computer, a software based modem and WebTV that made him famous. I also learned that, in a startup, the founders should get along and they should have a common visions, because if they don't get along, there will be so many problems that they won't be able to face and I think the company won't be able to survive it that happens.

CASE STUDY #14

TiVo

TiVo is a digital video recorder wherein it gives the users the power to manipulate television. With TiVo, viewers could skip commercials, pause live TV, and schedule the recording of every episode of a series. TiVo originated from a complex idea which was to create a network server for homes. This idea was thought by Mike Ramsay and Jim Barton, who cofounded TiVo. But they realize that it would be hard to explain to consumers why they need it, that's why they thought of much more simpler idea and that's where TiVo came in. Mike Ramsay thought that nobody had really thought that viewers could record television on it.

TiVo was launched at the end of March of 1999. They called it the Blue Moon event because it turns out that the month of March was a blue moon. After they shipped their product, they declared that day a company holiday which is still being practice until now.

After a few months, they had an argument with Replay, a company who is also doing the same thing as TiVo, wherein Replay was claiming that they were the first company who released the product and shipped it first. But the truth was that TiVo was actually the first company who released the product and shipped it first.

Mike Ramsay also had an arrangement with Philips. I think TiVo is a fortunate company because the press loved their startup and that they were screwing up with the media industry. And also, the press loved the idea that the TiVo Company was able to defend itself from Replay. Because of that, they were able to get huge amount of publicity.

They were also able to have a deal with Sony. Not only that, but they also have an important deal with DIRECTV. TiVo started supplying DIRECTV with TiVo's.
The deal with DIRECTV was a big thing for Mike Ramsay, because of DIRECTV, TiVo grew and they had already branch out to other partnerships. I think that thing you can actually pause a live TV made their product a big hit.

In this story, I learned that if you're starting a company, you should have low expectations but high ambitions. You should have low expectations because you have in a startup there are a lot of barriers you have to cross before you can really say that you have a company and you've got this thing running. You should also think whether this is the right thing you should do.

In a startup, it's very important that people should think through what they're trying to get done, not just the technology standpoint but also the business standpoint. And also, you should be prepared to stand up and justify this as rationally as you can. Also, you should never doubt that you are going to succeed you just have to think positive and that everything is going to be alright.

When it comes to the founders, you have to have a strong bond and a synchronized vision of what you are planning to do and what you want to achieve. Having a strong bond and same vision will make your company better. And also it will make you succeed in your plans.

CASE STUDY # 15

Viaweb

Viaweb is a web-based online store wherein the software is running on the server and lets the user to control it through their browser. Viaweb was one of the first companies to create an online store. The Yahoo Store that we all know we originally Viaweb. Viaweb was acquired by Yahoo in June 1998 and they renamed it as Yahoo Store.

The creator of Viaweb was Paul Graham. Before Paul Graham founded Viaweb, he had a different startup before Viaweb and it was Artix. Artix is an online art gallery. Artix didn't become successful because the artist didn't want to place their arts online.

Paul Graham tries to convince them to put it online but they still resist it. So Paul Graham thought that maybe they should do something that people actually want. That's when online stores come up.

The name Viaweb came from their idea of online store. Viaweb worked via web. At first, they thought of writing software wherein the users could use the desktop computer to build a website then they will upload it to a server. After a couple of days, they thought why don't we run this on the server and then the users can control it just by clicking on the links on the web page. They were able to make it work and they felt pretty excited because they were able to build a program without having to learn Windows because they don't know how to write Windows software and they don't want to learn it. Another reason why they felt so excited is because now that they were able to make the software run, so that means that they could start a company.

They got their first funding from Julian, who also worked with them on Artix. He gave them $10000. After 6 weeks, they are getting more and more busy. So they got Trevor Blackwell, who was the smartest grad student in the computer science program, to work on it too. Before they got Trevor Blackwell to work with them, Paul and his friend Robert had an argument over the phone. Robert was saying that they're already working on this software for a whole month and still it's not finished. So Robert basically rebelled, that's when Trevor came to the scene. After they finish the demo of the first version, they didn't get the money from the investors because the investors wanted a majority share of the company for a small amount of money. They try to improve it and they demoed it for the second time. This time, they got $100000 from angel investors. After that, they wrote more software's and finally they started trying to get users. Their first customers were a pair of technical bookstores.

One of the biggest turning point that they had was when a big company called them and said that they to buy them. But the deal didn't go nicely. During the meeting, Paul asked one of the investors to be the negotiator and negotiate with the big company. The guys from the big company offered them $3 million. But the investor said, "Well, I won't sell you the company for $3 million, but for $1 million, I'll sell you an option to buy the company in 6 months for $20 million." At that time, the guys from the big company got pissed off so they just walked away. Paul Graham felt miserable about what happen and tried to ask the guys from the big company of they're still thinking of buying the company. The big company decided not to buy them anymore. After all the problems they have faced. Finally, they were acquired by Yahoo.

This story tells us 3 things. First, make the software as simple as possible so that the users won't have a hard time using it but powerful enough to amaze them. Second, it's never a deal till the money's in the bank. There are so many things that might happen during the deal so don't be too confident if somebody wants to have a deal with you. Third, "Don't judge a book by its cover." Don't estimate people who don't seem very businesslike and don't seem very well dressed. Don't just look at the outside appearance but also the inside appearance because what's more important is that things they've done for the people.

CASE STUDY # 16

Del.icio.us

Just like any other startups, del.icio.us was built because Joshua Schachter built it for himself. He needed a way of organizing his collection of 20000 bookmarks and he thought of the idea of tagging them with brief explanation to help him find links easily. Del.icio.us became popular because of spread of mouth. This startup started from a hobby then turn into a company.

Back in 1998, Joshua Schachter had this website called Memepool. Users would email Schachter good links and he had to look at it one by one and write it down, after that, he will then post it in Memepool. It took him long time to post everything because he's not a good writer so there came a time where he was piled up of links. There was a time in 2001, he had a text file filled with 20000 links. That is where tagging came in, Schachter started putting in notes. He would put the URL then a word or two describing it. After putting all in notes, then he will group the things together.

After all those work, he decided to build another website called Muxway, in 2001. Muxway was a bookmarklet; you can save things; you could describe and tag them. It was single-player and no one else could use it other than you but the actual website was visible to other people. After some time there were already 1000 readers looking at Schachter's stuffs. And more and more people are subscribing to his bookmark.

In the late year of 2003, Schachter decided to build another website called del.icio.us. This is somewhat combination of Muxway and Memepool. Just through the spread of mouth, del.icio.us had already 30000 users by the end of 2004. Joshua Schachter is one of the founders who wouldn't accept huge amount of capital. Maybe because of if you have a huge amount of capital, you might be overwhelmed by it because you can buy or do anything you want. And if that happens, the money will become less easily. So Joshua Schachter just want a small amount of capital then from that money, he would then work on it for a while and see if it turned into something then see where this goes.

This story tells us 3 things. First, there are always 2 stories going on simultaneously in any startup. One is the software story, you can write as much great software as you want and more and more users are loving it, second is the business story, business story is always the turning point in every startup. Even if you have great software and many users, it's useless if you don't know how to run your business, if you make one wrong decision, all your hard work might all gone in just a snap of a finger.

Second lesson is that try to have a CEO who is willing to be the COO and do the business stuff and let you do the technical stuff. You don't need a CEO who's going to tell you what you should do even if that CEO doesn't have any knowledge about computer stuff. Lastly, make something that will make people happy and your business will grow.

CASE STUDY # 17

ONElist & Bloglines

ONElist and Bloglines have one founder and that was Mark Fletcher. Before Fletcher started ONElist, he was the senior software engineer in Sun Microsystems. ONElist was a free Internet email list service.

He started ONElist because he wanted to start a mailing list for his parents, because during that time, before you could put together a mailing list, you should download software and you should have a computer connected to the internet. For an average user, this is a hard thing to do. So Fletcher thought of a free internet mail listing service. Fletcher started ONElist in 1993 then after 3 years Yahoo acquired ONElist and renamed it as eGroups in June 2000.

After ONElist was acquired by Yahoo, Fletcher leaves the company after the acquisition. 3 years after ONElist was acquired, Fletcher started another startup called Bloglines was self-funded and Fletcher didn't deal with any investors.

After 2 years, Bloglines was acquired by Ask Jeeves in February 2005. When Bloglines was acquired by Ask Jeeves, Fletcher didn't leave the company unlike in ONElist. He stayed for 14 months. He helped building up a team an make sure that their reputation will still be good after 5 years or more. He also made sure that the knowledge in his mind will be transferred to all the programmers in the team. After that, he left the company with confidence that the company wouldn't fall apart.

This story tells us 2 things. First, business is a learning process. You learn everything in your previous startups. When starting a new startup, you should apply all of what you've learn from the previous to your present startup. Don't do the same mistake twice.

Always try something new and different; you don't know what might happen. Maybe that thing might change your life as well as everybody's life. Just like on ONElist, Fletcher did deal with the VC's, but on Bloglines he just self-funded it. He just tries out what will happen if he didn't take the VC on Bloglines. And also, Fletcher just solved his own problem first and then it became business.

Second, do what the users want. Like in Fletcher's startup, users send him emails saying, "This is what I want." He's getting 50 to 100 emails a day at Bloglines and most of them were suggestion about adding features. If you are taking action with their suggestions and if you users see that you are actually listening to their suggestions they will become loyal to your site and because they felt that they are able to participate and express their thoughts. And also they felt that they are being valued so that's why Bloglines become successful.

CASE STUDY # 18

Craigslist

In 1995, Craig Newmark, founder of craigslist, started a hobby which was an email list to publicize events in San Francisco. Craig Newmark started sending out notices about cool events to his friends. Then, as time passed, more and more people wanted to be added to the list. And the users started calling it “Craig’s List”. The users started suggesting other kinds of things, like jobs or stuff for sale. The craigslist became so popular in just a short time. Both subscribers and the number of postings grew quickly.

Craig Newmark was so surprised because users are starting to use the mailing list for non-event postings. Users demand for more categories so the list of categories grows. Because of popularity, Craig Newmark thought of switching the mailing list into a website and adds categories. Craig Newmark realized that the site was growing so fast so he decide that he should stop working as a software engineer and work full time running the craigslist. Over time, craigslist became a central network of online communities, featuring free classified advertisements and forums on various topics.

At first, craigslist was just a side project. Craig Newmark would be doing his contracting work first and get stuff done then he would take a half hour off to do craigslist then he would get back to work. When craigslist became popular, many venture capitalists want to fund them. At first, he wanted to accept the offer but Craig Newmark basically refuse to take the offer. He refused to take the offer because he had already made enough money to cover the costs. And during that time, they were charging less than 1 percent of the site.

Craig Newmark funded craigslist all by himself. He didn’t even take any outside investment, because they were trying to run a non-profit company and the non-profit entity only took a few loans. So Craig Newmark was able to fund it just by himself all this time. There are also companies who also wanted to buy them but Craig Newmark still didn’t accept the offer.

At first, Craig Newmark was having trouble in managing things but when he got Jim Buckmaster to be the CEO and him as the customer service rep and founder, everything went smoothly.

Why did people love craigslist? Why craigslist did become popular? This is because craigslist gives people a voice and a sense of community trust. Other factors to be considered is that craigslist have down-to-earth values, customer service and simplicity. Their site created a culture of trust and fairness to other people. And their site helps the users to get everyday work done easily.

People’s suggestions and feedback are very important because that’s how your company will grow. Just like what Craig Newmark did when people would suggest things to him, he would then think if that suggestion is what most people would like then he will do it for the sake of the users. Most of the functions that can be found in Craigslist were

based on the suggestions from users. People felt that they are being valued by Craigslist so they are loyal to the site that's why Craigslist become so popular.

CASE STUDY # 19

Flickr

As we all know, Flickr is an image and video hosting website, web services suite and online community platform. Flickr was just a feature that Caterina Fake thought of. Caterina Fake started Ludicorp with her husband, Stewart Butterfield, and their friend Jason Classon.

Ludicorp started when Caterina Fake and her husband, Stewart Butterfield came back from their honeymoon. Ludicorp came from the Latin word ludus which means play. So they thought of building a massively multiplayer online game which was called Game Neverending. This game is based around social interactions, players could form groups, instant message each other and there was a social network in it.

The company's first product, Game Neverending, is a web-based massively multiplayer online game with real-time interaction. As time passed, they thought of adding a new feature, a chat environment that includes photo sharing. That's how Flickr was able to surpassed Game Neverending. Flickr was just a service emerged out of tools originally created for Game Neverending. But Caterina Fake and her team thought that Flickr could be something big so they decided to stop the development of Game Neverending and they focus more on a new photo-sharing community site.

At first, Flickr started off as a feature. It wasn't a real product. It mas a kind of instant messaging in which users could drag and drop photos onto people's computer and show the users what are you looking at. In just 8 weeks, they were able to build the feature and presented it at the O'Reilly Emerging Tech Conference. It did have positive reactions from the people but it's not a compelling product because it was just a feature. The feature was useless unless you are really in it. But more importantly, this feature grew popularity.
It started to get traction when they added the ability to put pictures on a webpage. As Flickr was growing, they tried to do both Flickr and Game Neverending at the same time. It was a tough one because there were only 6 people in the company and that just wasn't enough resources to do both. There are so many people who signed up to test the Game Neverending but Caterina Fake doesn't have any choice but to put the game on hold and stop the development because Flickr was growing rapidly.

One thing that is unique about Flickr is that Flickr was used as a news-update through pictures. This what makes Flickr so popular. These days, people could also upload photos to Flickr without using a computer. If something happens people can take pictures using their cellphones and upload it immediately to Flickr. That's how the news-update in Flickr runs. There is also a tagging feature so people could find things faster.

In the business world, sometimes you expected to do something but you ended up doing a different thing. Sometimes what you have planned won't really happen. Just like what happen to Ludicorp, at first they were developing Game Neverending but they ended up launching Flickr. So you don't really know what might happen during the development of a product. One more thing about this story is that, there was a female founder. This wasn't an easy job because some people could criticize the ability of women. And Caterina Fake proved them wrong. There is no difference whether you're a male or female founder. The important thing is that you have determination and passion for the business.

CASE STUDY # 20

WAIS & Alexa Internet

Brewster Kahle was the founder of WAIS (Wide Area Information Servers) while he was working at Thinking Machines. In 1993, he left Thinking Machine and focus WAIS Inc. WAIS was one the earliest forms of Internet search software. During the first few years of WAIS, they don't have any investment at all; even Brewster Kahle himself had no savings so it was self-funding from the start.

Brewster Kahle got the idea of WAIS when Thinking Machine was trying to figure out "How would make use of a machine that has 15 gigabytes of disk space and would have processors that you could ran at a gigahertz?" after figuring out how to do it, Brewster Kahle prototyped it during his free time. He was doing WAIS while he was in Thinking Machine.

After Brewster Kahle left Thinking Machines, there were some questions like whether Thinking Machine owned the idea, because first of all Brewster Kahle started WAIS while he was at Thinking Machine. So there might be some ideas that he got from Thinking Machine, that's why they thought that maybe Thinking Machine owned the idea of WAIS. But Brewster Kahle made sure that all the things that WAIS needed was based on public domain of software that had been produced and it was based on the open source software. So that means that there were no copyrights. Fortunately, they were able to solve the problem.

After 2 years Brewster Kahle sold WAIS to AOL in 1995. After the acquisition, he didn't leave the company immediately, he tried to work within AOL but it was very difficult.

After a year, Brewster Kahle left the company and started another company with Bruce Gilliat. This was Alexa Internet or Library of Alexandria. Alexa internet is best known for operating a website that provides information on web traffic to other websites. And also, Brewster Kahle start another company called Internet Archive. So Brewster Kahle started both companies simultaneously.

Interest Archive is a nonprofit organization which aims to maintain an online library and archive of web and multimedia resources. Alexa Internet offers a toolbar that

gave the users guidance on where to go next, based on the traffic patterns of its user community. The use of Alexa Internet is that everything that Alexa collected would be donated to the Interest Archive. In 1999, Alexa Internet was acquired by Amazon, and he continues to run the Internet Archive.

This story tells us that finding a partner is difficult. Finding a partner is like finding the person you want to get married with. To find a good partner, you should have trust on each other, even through hard times you should always help another. Second, in achieving your goal, you should have a plan on how to accomplish it. Just like what Brewster Kahle did, he makes sure that every year there's some accomplishment on his plans. We just have to strive harder in order to achieve our goals.

CASE STUDY # 21

Adobe System

Charles Geschke and John Warnock founded Adobe System. Adobe System was founded when Charles Geschke and John Warnock was still working at Xerox. They developed a language called Interpress that allows any computer to talk to any printer. After developing Interpress, they showed it to the Xerox management and the Xerox management was excited about it. The only problem they have was that it would at least take them 7 years to bring a product out. That's the fastest way Xerox could get a product out. Charles Geschke and John Warnock thought that if they will wait for 7 years before they could get the product out, maybe by that time users won't be interested anymore in their product. So finally, they decided to bring their ideas and start a startup.

Their first product was PostScript. PostScript helps describe complex documents into a simple form. One problem they encountered was when Steve Jobs left the company and Jean-Louis Gassee replaced him. That's the time when Jean-Louis Gassee and Adobe started not to get along. Jean-Louis Gassee was tired of paying Adobe royalties for the Laser Writer and she thought that they don't have to pay them anymore. This problem was solve when Apple realized that it took Microsoft and Apple 3 years before they could shipped a product while Adobe was able to sold hundreds of thousands of units in just the first year. So Apple decided to just redid the deal in PostScript with Adobe.

One of the major turning points of the company was when IBM and HP realize that the PostScript was so popular so they decided to get business deals with Adobe System. Another turning point of the company was when they develop Illustrator. John Warnock's wife wanted to get her concepts out on Laser Writer. So John Warnock builds a program that acts like what a graphic artist would expect to have. So that's where Illustrator came from.

For me, the products of Adobe System a big hit to the users because it was very useful for the users. The PostScript made the lives of the users easier. I think in the history of Internet, Adobe System gives great impact to the users because they were recognized as industry leader in graphic and desktop publishing software. And until now, many users are using this software.

One thing that is unique in this startup is that their service was never offered before. I think they were the first one to develop Photoshop, Illustrator and Acrobat application. If not, I guess their product is much more popular than any other application.

If I were in the co-founders shoes maybe I would have done the same thing, but I'm not sure of it because I'm not actually in their position and as of now, I don't have enough knowledge to decide what should be done and not.

I think startup would be possible in a third world country, if the people will just think more creatively; there is a possibility for that to happen. Just like in the Philippines, Filipino people are very creative especially when it comes to inventions and graphic arts. Maybe there will come a time that Filipinos would start a startup that is more unique than Adobe.

Before I read the case study, I thought that it would only take a few months for a company to get a product out but after reading the case study, I was shocked that back then, it took them at least 7 years just to get the product out.

CASE STUDY # 22

Open System

Ann Winblad started Open System with the help of $500 she borrowed from her younger brother. Open System is an accounting software company. Open System started when the group of Ann Winblad was chosen under the Request of Proposal bid to build a student accounting system for a vocational school. So that's when they started programming accounting systems for smaller computers.

One of the major turning points of the company was when Ann Winblad was able to get 12 $10000 checks from the CADO computer guys. Ann Winblad was telling them that if they give her a check worth of $10000 today, the guys from CADO could have unlimited rights to one of the modules which was the general ledger. Because Open System was a core component of the company so CADO wanted Open System to be successful so that all the software would work in his resellers.

After 6 years, Ann Winblad sold Open System for $15 million. Then she started another startup called Hummer Winblad Venture Partners. Hummer Winblad Venture Partnerswas the first venture firm to focus exclusively on software. For me, Open System is not that popular, it is useful for students who take accounting courses and people who use accounting in their work. Limited people could use this system so I think it's not popular worldwide.

For me, Open system and Hummer Winblad Venture Partners didn't have great impact in the history of Internet unlike other companies like Microsoft, Apple, Google and other big companies. Maybe because one thing is that I haven't heard of their company and I think that not everyone could use their system. But their system did help a

lot for people who know accounting. For them Open System did have a great impact in the history of Internet.

One thing that is unique in this startup is that their founder was a woman. As we all know there are only few founders in a startup that is woman. I admire her for her strengths, determination and skills. Back then, she was just an ordinary girl who was having problems regarding money. She was very lucky because she was picked as "experimental" student. So she doesn't have to take any pre-requisite like most of the student. And she was able to get double major in mathematics and business administration. She also takes some computer science classes, that's how she was able to build a system. If I were in her place, I don't think I could do all those things. It's hard to take up double major then having computer science classes. But I think that's the reason for her success. She didn't give up; she kept fighting until she reaches her goal. Look at her now; she is the most powerful woman in venture capital.

I think this startup could also happen in a third world country as long as you have the determination, strengths and skills to do it. It's not easy but we just have to try and try, you don't know, maybe it would be also successful just like them.

Before I read the case study, I didn't know that there was a startup called Open System and Hummer Winblad Venture Partners. After reading the case study, I learned that Open system was an accounting software company while Hummer Winblad Venture Partners was the first venture firm to focus exclusively on software. When I searched for information at Wikipedia, I realized that it wasn't a popular company so there was no information about it.

CASE STUDY # 23

37signals

The founder of 37signals was Jason Fried. Before, 37signals was just a web design shop. But David Heinemeier Hansson was the one who helped them transform 37signals from a consulting company into a product company. Then he wrote the first product of 37signals which was Basecamp. Basecamp is an online project management tool. He was also the one who developed Ruby on Rails, a free web application framework, and Instiki Wiki, a Ruby-based Wiki clone.

One of the major turning points of the Basecamp was that, Basecamp was very general in project management. It was not specific like billing and time tracking. In short, whatever project management that people want to do, they can do it on Basecamp. If you compare Open System with Basecamp, Basecamp is more useful than Open System because Basecamp is very general in managing projects; users can mange different kinds of projects, from simple projects to complex ones. Users can do everything they want. While in Open System, they are very specific so there are only few users who use their system unlike Basecamp, there are so many users who loved their product.

Some of the mistakes that they did was that, they didn't figure out that bank wouldn't let the company to bill people once a year. So they change the plan from billing people once a year to billing people every month. Another mistake they did was that they didn't consider the time-zones. They just assume that everyone is in the Central Standard Time. So they have to fix again in order for other countries to participate.

For me, people used the Basecamp because of course it does help them and it makes their life easier. It uses less time in managing plans than manually managing it. I think people also love it because it can do all kinds of stuff with it like managing wedding, home improvement or even for personal use. At first, they just do it for the company's use. But after getting feedback from the users, they too have the same problem as theirs, so they launched it so not only the company would benefit from it but also for the users too.

I think Basecamp can influence other startup company by providing something that will not only benefit the company but also the users. Maybe other startups might be influenced in creating something that the users want. For me, most startups are the same because they are trying to do something different and unique. In Basecamp, I think the unique thing about the product is that people could manage their projects online. If I were in their shoes, maybe I would check all the possible mistakes that we could have done before launching out product. Once I'm assured that all mistakes have been corrected, that's the time when I'm going to launch it so it would go smoothly.

Everything is possible anywhere. This startup or product can also be possible in a third world country. Developers just have to think something that will be useful for the people and that the users would love it.

Before I read this case study, I didn't know everything about Basecamp or 37sisgnals. After reading the case study, I learned that Basecamp is an online project management tool and after I checked it out, I realized that this is really a cool site and very useful to people.

CASE STUDY # 24

ArsDigita

The founder of ArsDigita was Philip Greenspun. ArsDigita is a famous startup in the world both as the embodiment of a new model for software consulting and as an all-too-colorful example of the dangers of VC. As of now, after ArsDigita went down in 2002, Philip Greemspun focuses now on his passion for photography and being a helicopter instructor.

The reason of the formation of this startup happened when Philip Greenspun went to Alaska for his photo blog project. He was documenting all his travels then he will email it to his family and friends. He also published his photo on the web page. After a while, he started getting questions from users asking questions regarding photography. He then will answer all the questions one-by-one. So he decided to make a forum on his

server so that there are users who have the same question, he will just answer one of them and they will both het an answer. And sometimes other users answer those questions for him. So that's where the photography community site came from.

After that he decided to write software to run his community site and he decided to post the code as an open source project. Then he started receiving calls from big companies that wanted to use the software but it needs some customizations. So that's when he and his friends decided to start a startup and that's where ArsDigita came from.

One of the big problems they encountered was about the venture capitalists. ArsDigita was profitable with $20 million in yearly revenue and they had $3 million in profits. But the thing is that they don't have venture capitalist. So the underwriters won't take them to public unless they have a big name venture capitalist like Kleiner Perkins.

The underwriters don't want to do research in a company. They just wanted to know who are their venture capitalists, then they will decided if they will take the company to public or not. For me, I think people used the services of ArsDigita because this caught their interest. It's something that they like to interact with. One of the unique things about ArsDigita was about their marketing and sales. It's usual for us that if we advertise our products we use ads but ArsDigita was different, their marketing and sales was educational. They will teach people stuff and maybe some of the people they taught will become their customers.

For me, if I was in a startup just like ArsDigita, I think I will focus more on getting really good people who shared the company's vision. I won't focus more on competition because I know that we are different from other companies and that we are unique. And I have faith in my company. I think if focus on competition t might get more problems and that could cause my company to fall.

As a conclusion, after I read this case study, I realized that there are so many biases in the business world. One is that underwriters won't take a company to public unless the company has a venture capitalist. They don't even check what does the company does. That's why some of the startup fall because they were not checked carefully, as long as you have venture capitalist there will be no problem, but that's wrong. Before a company can go public they should check what do they do, they should have a background checked first. It's not important of the company has venture capitalist, as long as it will be successful in the future then it is good for them to go public.

CASE STUDY # 25

Fog Creek Software

The founders of Fog Creek Software were Joel Spolsky and Michael Pryor. Joel Spolsky was inspired by ArsDigita. He wanted to start a startup just like what Philip Greenspun did. He also wanted to have same values as Philip Greenspun regarding programmers, that they wanted to be programmer-centric and a place where they wanted to work.

At first, they don't have any idea on what they are going to do. They just have any idea that if those punk kids who just graduated from college and knew nothing about anything were able to go public with $100 million valuations, surely they can do much well than those kids do.

After Joel Spolsky started a startup, he started writing Joel on Software to share his thoughts about software development, management, business and internet. This was his programming blog. And this was his strategy for software startups. He wanted to create a popular blog to get attention. He founded Fog Creek Software in September 2000 but sadly his consulting bus collapsed by November 2000.

The problem they encountered was that when the company was growing, they will hire consultants to help the company grow more quickly, but when the company is shrinking, they will immediately fire all consultants. Their market was going down by 0.002 percent instead of growing, so they decided to first fire all consultants. That's why their consulting business totally collapsed.

Luckily, Joel Creek Software hadn't grown very much and it didn't have much consulting business to lose so they were able to survive it. After the consulting business collapsed, they now focused themselves on producing an internal bug tracking application that became their first product which was Fog Bugz.

Even if their original plan didn't work they were still able to recover from the fall of their consulting business. After they shipped Fog Bugz, they saw a small yet steady-growing sale each month. They started earning $5000-$10000 a month selling Fog Bugz. This was enough to cover all their expenses.

Joel Spolsky had this unique idea that he wanted to have a big consulting company and then he will build a software company inside the consulting company. Although he wasn't able to gain accomplish his plan, but he was able to gain experience from it and next time that he tries to do it again, maybe he will be able to succeed. I think Fog Creek Software won't exist it ArsDigita and Philip Greenspun didn't exist, because Joel Spolsky got his ideas and inspiration from ArsDigita and Philip Greenspun.

I think Joel Greenspun can still accomplish his idea of building a consulting company and a software company inside it, as long as he believes that he can do everything and as long as he focuses on it and he put his heart in it. After reading this case study, I realize that ArsDigita was able to inspire other people to start a startup that has the same values as they had despite what happen to their company.

CASE STUDY # 26

TripAdvisor

TripAdvisor is an online travel site where users can get reviews of destinations, hotels and attractions. This startup was founded by Stephen Kaufer, Langley Steinert, Nick Shanny and Thomas Palka. TripAdvisor started because Stephen Kaufer was sick and tired of searching the internet for information regarding the destination they were going.

Stephen Kaufer and his wife, Caroline, were trying to find a vacation for themselves. A travel agent would recommend them to a place, and then Stephen Kaufer will then use the internet to find out more about the place. There were lots of website that would help then book a reservation but none of it tells them whether the place was good or not. This happened not just once but twice. That's when his wife suggested that they could just build a better search engine to find what users are looking for in travel.

They had this idea of online travel site where there are unbiased opinions about a place or a location. They didn't immediately work on the idea; instead they put it aside first because he was still employed by that time. Then the time came that he decided that it's time to pursue his ideas. That's when they started TripAdvisor.

People liked their product because TripAdvisor was not a bias travel site. They were posting opinions of people whether it is good for the place or not. Sometimes, some hotels wants to sue TripAdvisor because TripAdvisor were posting reviews wherein it would put a bad shape for their hotel and that forces them to hsape up their company and deliver better service to stay in business.

There are times that the owner of the company is the one making a review of their company so their company would be in a good shape. The owner would pretend that he/she was one of the customers. TripAdvisor was able to stop this so it didn't become a big problem.

One unique thing about this startup was that even if it's hard to collect information about places, they still did it even if it took them a long time before they were able to get all the information they need. Somebody already told them that they're crazy because it's an impossibly a large task. It took them few years to finish. Now, they have an easy time in keeping current records with all the travel publication and the net result was that they had a small yet highly relevant database of travel information.

One of the turning points they had was when they got the help of Expedia. TripAdvisor wants to have ads so that they would earn extra cash. They offered Expedia service. TripAdvisor would just deep link to a reservation page on the site of Expedia and take a cut of the booking. It would be an easy job for Expedia in finding customers to book to them because TripAdvisor is already doing the job for them. TripAdvisor asked Expedia how it was going. Expedia told them it was great, so TripAdvisor started to ask payments afterwards.

Before I read this case study, I don't know that there was an online travel site. After readingthe case study, I learned that TripAdvisor is the largest online travel community in the world. Stephen Kaufer was able to achieve his goal of being the most popular travel site in the world because of his determination and he didn't think of quitting even if he is already facing so many problems with the company.

CASE STUDY # 27

Hot or Not

Hot or Not is a website wherein you can post a photo and have other people vote whether they are hot or not on a scale of 1 to 10. James Hong and Jim Young started Hot or Not. Before they started Hot or Not, they were already working on a website called XMethods. XMethods was the first directory of publicy available web services.

After XMethod, they started Hot or Not. This all happened when James Hong and Jim Young were hanging out drinking, and then Jim Young mentioned that he met a girl at a party and that she was a perfect 10. After that event, that's when James Hong thought that what if they had a service where other people could post their pictures and let other people vote them from 1 to 10.

So they worked on their idea, before they launched it, James Hong's dad saw the Hot or Not website and then he got addicted with it. His dad started commenting things like, "She's hot. This one's not hot at all." So they thought that their website would somehow be a big hit.

After launching it with their own pictures, James Hong started emailed it to his friends. After that, he got 40000 hits that same day. I think that was great for just a day and it was just the first day of the website. James Hong also tried to spread it to random guy outside. After a while, more and more hits are coming in from different people.

One of the problems they encountered was getting rid of the huge bandwidth caused by the pictures. So they tried to use Yahoo Geocities to serve the images and reduce the bandwidth that cost them $150K/yr. Another problem they encountered was their interview with Salon.com. Jim Young needed a server for XMethods so they decided to grab a 400 MHz Celeron box, drove it back to Berkely and plugged it into a corner in Jim's office concealing it under a pile of books. Around 5am, they already had the site back on and just in time to withstand the traffic from the Salon article.

After those two problems, Hot or Not was still surviving, but the site was still slow. So they were searching for solutions for running it on a managed server. They were able to talked to Rackspace into a deal where Hot or Not would host it for free and use Hot or Not as a poster child to prove their ability to scale and handle load.

One more problem they encountered was about the naked photos on their site. Hot or Not asked help from the founder of Double Click. Unfortunately, the first image that

the Double Click guy saw was a picture of a naked woman. Double Click won't help Hot or Notunless they are proven not guilty. So they build the moderation system in order to get advertising.

The system they developed to prevent these naked pictures was the "Meet Me" system. This is also a double-match verification way of linking two people. Then they started charging $6/month for using the system and they discovered that people would pay for it as a dating service.

For me, Hot or Not is a website that can be visited for people who are bored in the internet. They could use it for fun. I think it's a cool website too because there are a lot of people posting photos in it and many people are rating it. One thing I learned is that, nothing goes according to plan. So there's no use of planning because in the middle of the process, your plan might change.

CASE STUDY # 28

Tickle

Tickle is a site that allows user to learn more about themselves by taking various personality assessments online. This startup was founded by James Currier in 1999. This idea came to James Currier when he was in his MBA class. There was a Meyers-Briggs personality test at Harvard. After answering it he noticed that the test generates more conversation than any book or movie ever had among the MBA students. So he thought of building a site for personality test, because he thought that the data that he will be collecting would likely be accurate and everyone would naturally answer the test honestly because if they're taking the time to complete it then that means that they wanted to find out truthful results about themselves.

There were so many kinds of test in Tickle like parenting, relationship, IQ and communication tests. But the problem was that no one checked their site. So James Currier thought of launching a dog test, because he remember that advertising agencies says that if you want people to remember an ad, include babies and puppies. So that's where his idea of dog test came from. This test lets user to find out what kind of dog are they. 8 days after launching it, they already got a million people trying to enter their site. Once that they got many users using their and that their site is going down every 10 minutes, that makes an easier job for them to find VC's for funding.

I think for me, Tickle became popular because most people were amused by their site. Their site is like a relaxation for some people. This site also helps people to learn more about themselves. Sometimes people use this site as a past time when they don't have anything to do.

If I were at James Currier's shoes, I think I will also do the same things that he does. His startup went smoothly, well there were some problems that he encountered but it went well. He already got millions of user just because of the dog test. And he was able to get VC's to fund them easily. Once they got funding from VC's, people would start

wanting to work for them, loan for them and even investment bankers want to meet them. Just because of a dog test he became successful. For me, in a startup, you should be smart enough to attract many user and VC's, that way you won't mind so many problems.

After 5 years, Tickle was acquired by Monster in 2004 for about $100 million. James Currier lets Monster to acquire Tickle than Yahoo because he felt that if he let Yahoo to acquire Tickle, they would have to move from Wall Street to Mountain View and Yahoo would make them change a feature on a division of a department. There would come a time that everyone would leave and the whole thing would just die, unlike in Monster, they managed to preserve the startup feel and keep doing their thing. Then after Tickle, James Currier founded another startup called Ooga Labs. Ooga Labs is a digital media studio that develops consumer Internet applications.

Before reading this case study, I didn't know that there was an online personality test. After reading this case study, I learned that Tickle was the largest career testing site in the world. I never thought that an online personality test would be this popular and that people would actually spend some time to take the tests.

CASE STUDY # 29

Firefox

Firefox is a web browser just like Microsoft's Internet Explorer. This started as a hobby side project and was never intended to become a business. Blake Ross and Dave Hyatt were the creators of Firefox. They started Firefox in 2002 because of their frustration from Netscape's browser. Before, Blake Ross had interned at Netscape, when he returned during summer, he found out that Netscape browser had become a frustrating one and they were sacrificing the user experience just to monetize the browser and become a promotional vehicle to support the Netscape.com portal site. So Blake Ross and Dave Hyatt thought of building a browser called Phoenix, which became Firefox, to solve the problem in Netscape.

Blake Ross and Dave Hyatt didn't feel financial pressure unlike any startups because money was just always sort of there. There were donations, seed money from AOL and they also got a deal with Google. So naturally they don't really have to worry about money, they got so many sources. Also, one thing is that Firefox was just a hobby, so they didn't care about money.

They didn't encounter any financial problem, but the only thing they were worried is about the name of their browser. At first, it was called Phoenix, but after a while they found out that there was also a company called Phoenix Technologies and that they were also doing a web browser. So they renamed it as Firebird, because of their logo. The problem was that there was already an open source database called Firebird, so they have to change the name again. Finally, they come up of the name Firefox which in Chinese means red panda.

Firefox didn't have any problem in getting users, but the thing was that the users that they were getting weren't their target audience. The people who are using Firefox were geek people. They were doing Firefox designed to be usable for a non-technical people. In order to get other people to use their browser, they build a site called SpreadFirefox.com so that Firefox will be spread through the word of mouth. During the first week of launching, they thought of contacting 100 different bloggers that had written about it and they also found out that most of the bloggers were willing to display a "Get Firefox" button on their site. Just because of Spread Firefox and bloggers, they got 250000 people to sign up. That was a huge success for them.

At first, Microsoft didn't mind about the competitors but when Firefox became a big hit they got alarmed so they tried to be the best browser. But I think Firefox is still much better than Internet Explore because Explorer was basically been abandoned and Microsoft already disbanded the Internet Explorer team. I also think that Firefox is much faster than Internet Explorer.

After Firefox, Blake Ross and Dave Hyatt started their next project which was Parakey. Before reading this case study, I didn't know that there were so many people using Firefox and I was thinking that Firefox and Internet Explorer were the same that they don't have any difference. But after reading this case study, I realized that Firefox is much better than Internet Explorer.

CASE STUDY # 30

Six Apart

Six Apart was founded by Mena and her husband, Ben Trott. This is a software company where Movable Type and Type Pad came from. The name Six Apart means the number of days between their birthdays. Six Apart's first product was Movable Type. Before Movable Type was created, Mena Trott was using an existing system which was a personal blog.

They created Movable Type because their existing blog, Dollarshort, was becoming popular and Mena Trott was not satisfied with the blogging software available during that time so they decided to create their own blog and share it with friends.

Movable Type is a professional blogging platform. At first they acted as consultants for awhile delivering customization on Movable Type. Then after Movable Type became popular, they decided being a consultant they rather just productize it and sell it to consumers.

After getting fund from Neoteny, they started another product called Type Pad. Type Pad is the world's premier blogging service, hosting the leading blogs and small business websites and powering the most influential voices on the social web.

Like any other startup and founders, Mena and Ben Trott lack experience especially when it comes to business things. But even if they lack experience they were

still able to go through difficult times because they have the strength and determination to face those problems. But their startup won't be successful without the help of their CEO who was Barak Berkowitz.

He helped coach them through some of the basics in setting the business strategy and doing the things that are needed to be done like establishing the basic operational infrastructure and negotiating a lease. Another thing about their CEO is that, Barak Berkowitz was willing to do everything that Six Apart needs and he wasn't an investor who would just gave them money and check them every quarter.

At first they got their funding thru donations. Even if they didn't ask for money people would actually give them money. What they did is just that they had one page up that said, "We take donations and this is why you should donate." I think people would actually give donations because they love their product and I think the second reason was that Movable Type gives recently updated keys. People would actually pay $20 just to get a key. So that's how they got their funds to pay the expenses.

Before reading this case study, I was thinking that if you are planning to start a startup you should know how the business runs and you should also have some experience with it. But after reading the case study, I realized that I was wrong, even if you lack experience you can still start a startup but you should find a CEO that would manage your business.

CASE STUDY # 31

Lycos

Lycos like Yahoo and Google is also a search engine and web portal that centered on broadband entertainment content. The founder of this startup was Bob Davis. Lycos began as a search engine research project by Michael Mauldin at Carnegie Mellon University in 1994.

Michael Mauldin was thinking that Lycos might be a big hit in the future but the problem is that he isn't a business person. So he decided to sell his project to Dan Nova of CMGI. They ended up starting a startup. Dan Wetherell, founder of CMGI's venture firm, bought 80 percent of the company and 20 percent of it was owned by Bob Davis and Michael Mauldin.

The first problem that Lycos had was about their work place. Their headquarters was based in Boston but Carnegie Mellon wanted them to keep a presence in Pittsburgh. So even if they have headquarters in Boston, they are obligated to have a meaningful presence in Pittsburgh.

The second problem they had was about staffing. During the first few months of Lycos, no one had ever heard about Lycos so they had a hard time in hiring people. Because people didn't know what is Lycos all about so nobody wanted to be part of their company back then.

Their first customers were their employees, their advertisers and their users. In order to promote their product, they focused on partnership with other advertising companies and getting other people to promote Lycos. The first 3 companies that were their licensees were AT&T, CompuServer and Prodigy.

After a few months, they tried to make some joint ventures overseas. They got somewhat partnership with Bertelsmann which was the largest media company in Europe. After that, they created Lycos Europe.

One of the things I learned in this case study was that, in an entrepreneur's life, there will always be setbacks, challenges, disappointments and failures. You can say that you are really successful once you were able to overcome the hardship and challenges that you are facing. That would determine that your business had succeeded.

I also realized that in a startup, everything is a complete revolution and there will always be new opportunities and new challenges that a founder will faced every single of his life. After a founder overcome one obstacle, there will always be another obstacle waiting. So in order to succeed you should a strong will and determination.

CASE STUDY # 32

Alliant Computer System & Shareholder.com

Alliant Computer System was founded by Ron Gruner, together with Craig Mundic and Rich McAndrew. This startup focuses on delivering high-end mainframes that could deliver more performance via parallel processing. Ron Gruner's goal was to be able to build a machine that uses multiprocessing to achieve a better performance than the fastest single-CPU machines.

The reason why Ron Gruner left Alliant Computer System was that he had an disagreement with some VCs. He's having problem with because some VCs are quite impatient with a difficult situation. Because Ron Gruner was having a financial problem with the VCs and he didn't like the strategy they have because he thinks that it was wrong, so he decided to leave the company. After a year, Alliant Computer System was bankrupt.

After a few months, he started another startup which was Shareholder.com. He became the sole founder of the company. But before he started this startup, he had 4 criteria in mind of what type of business he wanted to establish in the future. First, he wanted to build a business that has a recurring revenue stream. Second, he wanted to have no more than 2% of the total involvement consumed by the administrative overhead. Third, he wanted the company to be capable of being bootstrapped from a small amount of capital. Lastly, he wanted the company to be solely owned by him so as not to give up the power control if there would by another VC argument.

One of the biggest problems he encountered in shareholder.com was about the wrong hotline number that was given to a very large pharmaceutical company. If they didn't solve this problem fast, it would create a big mess and it might be the cause of the destruction of the company. So Ron Gruner hired a private investigator to search for the owner of the number 800. Once he found out who that was, Ron Gruner immediately called that person and told him the situation. Ron Gruner offered an exchange with him. He said that he would do anything just to acquire that number from that person. And that person agreed with the agreement. So Ron Gruner's problem was solved and that was a close one.

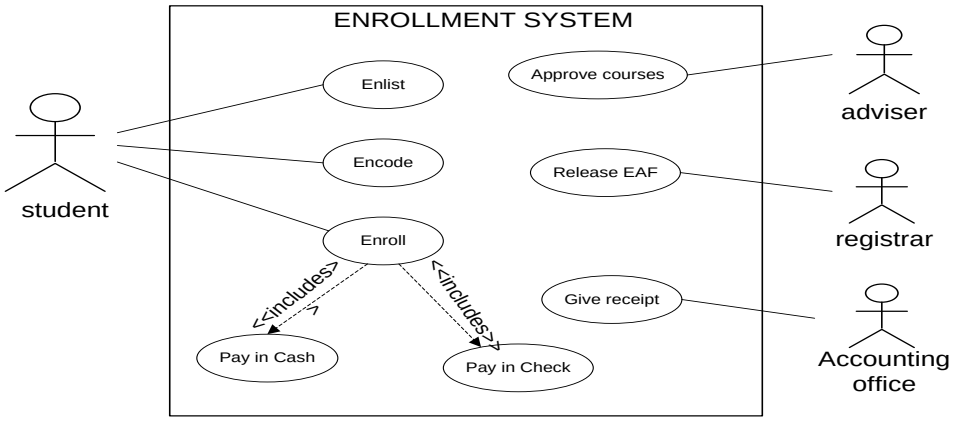
After 14 years of hardship, unlike the first startup which was Alliant Computer System that went bankruptcy after 10 years, shareholder.com is still alive and kicking, shareholder.com grew steadily and they were acquired by NASDAQ in February 2006.

This startup taught me about persistence, that we should be brave and continue to follow our dreams even if there are so many things that are blocking our way. These things are just obstacles that would test whether we really deserve to be successful in the future.

All in all Founders at Work taught me to have determination, perseverance and that we should be adaptable to all changes that might happen in the future. This also taught me that nothing goes according to plan that your plans might change in the middle of the process. Lastly, this book taught me that starting a startup is all about trial and error, that you can't be successful if you don't try new things and have a somewhat bad experience from it.

USE CASE

USE CASE # 1



Identification summary

Title: CSB Enrollment System
Summary: This use case allows a student to enroll in CSB.
Actors: student, adviser, registrar, accounting office
Creation Date: June 4, 2008
Version: 1.0
Date of Update: June 4, 2008
Person in charge: Lim, Marylyn Grace C.

Flow of Events

Precondition:

1. The computer is connected in the internet.
2. The student must have either the money or check.

Main Success Scenario:

1. The student log-in.
2. The student enlists subjects.
3. The adviser approved subjects.
4. The student encodes the schedule.
5. The registrar releases the EAF.
6. The student either pays in cash or check.
7. The personnel in accounting office give the receipt to the student.

Alternate Sequence:

A1: Student enters invalid id number or password.
A2: Disapproved subjects.

Error Sequence:

E1: encoding of schedule is closed.

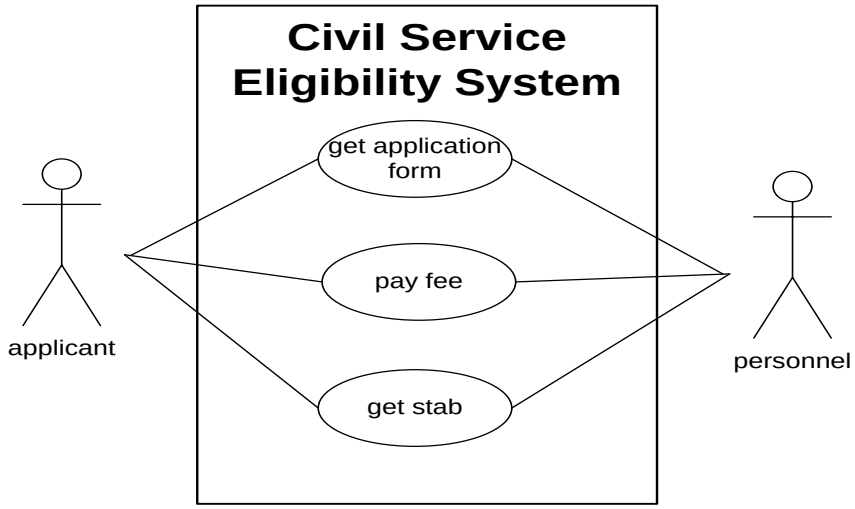
Post conditions:

1. Papers use for receipt is less.
2. Electric consumption.

UI requirements

1. Student's ID number and password.

USE CASE # 2



Identification summary

Title: Civil Service Eligibility System
Summary: This use case allows an applicant to get a civil service eligibility exam.
Actors: applicant, personnel
Creation Date: June 25, 2008
Version: 1.0
Date of Update: June 25, 2008
Person in charge: Lim, Marylyn Grace C.

Flow of Events

Precondition:

3. Applicant should have valid ID.
4. 4 pcs. ID picture
5. Applicant should have money.

Main Success Scenario:

1. The applicant gets an application form.
2. The applicant fills up the form.
3. The applicant gives requirements.
4. The applicant pays the fee.
5. The applicant gets the stab.

Alternate Sequence:

A1: from 2

3a. the applicant doesn't have complete requirements.
4a. the applicant completes the missing requirements.
5a. back to 3

A2: from 3

4b. applicant doesn't have enough money.
5b. applicant gets the exact amount of money.
6b. back to 4

Error Sequence:

E1: from 0

1a. office is closed.

2a. wait until office is open.

3a. back to 1

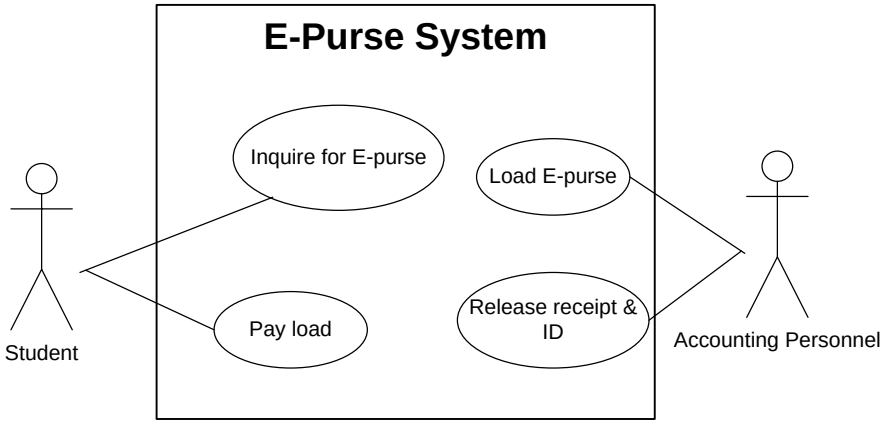
Post conditions:

3. Applicant can take the exam.
4. Applicant has less money.
5. Application form becomes less.

UI Requirements

1. Applicant's money
2. Applicant's valid ID
3. 4 pcs. ID picture

USE CASE # 3



Identification summary

Title: E-Purse System

Summary: This use case allows a student to put money into his/her E-Purse.

Actors: student, accounting personnel

Creation Date: June 18, 2008

Version: 1.0

Date of Update: June 18, 2008

Person in charge: Lim, Marylyn Grace C.

Flow of Events

Precondition:

1. Student is enrolled in DLS-CSB.

Main Success Scenario:

6. The student goes to window 2 to inquire for E-Purse.
7. The student gives his/her ID.
8. The student pays in window 1.
9. The student enters his/her password.
10. The student gets his/her receipt and ID at window 4.

Alternate Sequence:

A1: The student enters the incorrect password.

Error Sequence:

E1: The student's ID is damage.
E2: The student's ID is lost.

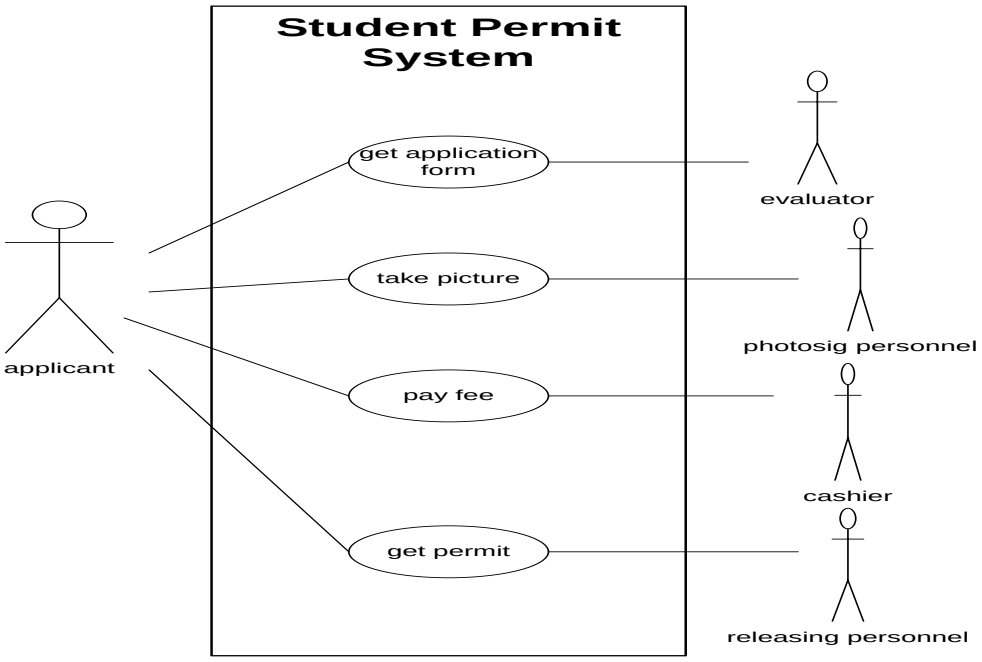
Post conditions:

1. Papers use for receipt is less.
2. Ink use in receipt is less.

UI Requirements

4. Student ID

USE CASE # 4



Identification summary

Title: Student Permit System
Summary: This use case allows an applicant to get student permit.
Actors: applicant, evaluator, photosig personnel, cashier, releasing personnel
Creation Date: July 2, 2008
Version: 1.0
Date of Update: July 2, 2008
Person in charge: Lim, Marylyn Grace C.

Flow of Events

Precondition:

1. Applicant should have a photocopy of his/her birth certificate.
2. Applicant should have a letter of consent.
3. Applicant should have money.
4. Applicant should at least be 16 years old.

Main Success Scenario:

1. The applicant goes to the customer service and gets an application form.

2. The applicant fills up the form.
3. The applicant goes to window 7 to pass the requirements.
4. The applicant goes to window 6 to take a picture.
5. The applicant pays the fee at window 5.
6. The applicant gets the permit at window 9.

Alternate Sequence:

A1: from 2

3a. the applicant doesn't have complete requirements.
4a. the applicant completes the missing requirements.
5a. back to 3

A2: from 4

4b. applicant doesn't have enough money.
5b. applicant gets the exact amount of money.
6b. back to 5

Error Sequence:

E1: from 0

1a. office is closed.
2a. wait until office is open.
3a. back to 1

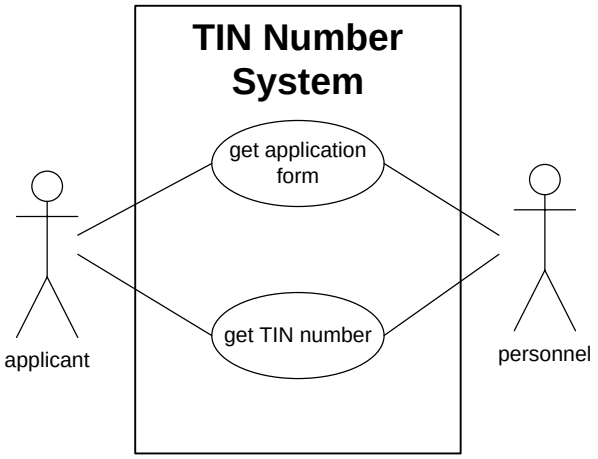
Post conditions:

1. Applicant has the permit.
2. Applicant has less money.
3. Application form becomes less.

UI Requirements

1. Applicant's birth certificate
2. Applicant's letter of consent
3. Applicant's money

USE CASE # 5



Identification summary

Title: TIN Number System
Summary: This use case allows an applicant to get TIN number.
Actors: applicant, personnel
Creation Date: July 9, 2008
Version: 1.0
Date of Update: July 9, 2008
Person in charge: Lim, Marylyn Grace C.

Flow of Events

Precondition:

1. Applicant should have birth certificate.

Main Success Scenario:

2. The applicant gets an application form.
3. The applicant fills up the form.
4. The applicant submits the application form as well as the requirement.
5. The applicant gets the TIN number.

Alternate Sequence:

A1: The applicant doesn't have complete requirements.
From 2
3a. the applicant completes the missing requirements.
4a. back to 3

Error Sequence:

E1: Office is closed.
From 0
1a. wait until office is open.
2a. back to 1

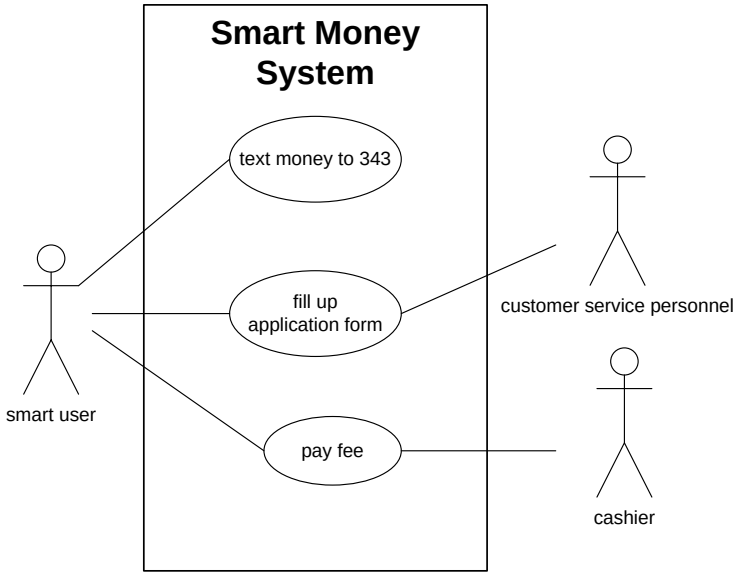
Post conditions:

1. Applicant has TIN number.
2. Application form becomes less.

UI Requirements

1. Applicant's birth certificate

USE CASE # 6



Identification summary

Title: Smart Money System
Summary: This use case allows a Smart subscriber to have a Smart Money Card.
Actors: Smart subscriber, Customer Service Personnel, Cashier
Creation Date: July 16, 2008
Version: 1.0
Date of Update: July 16, 2008
Person in charge: Lim, Marylyn Grace C.

Flow of Events

Precondition:

1. User must be a Smart subscriber.
2. User must have money.
3. User must have any valid ID.

Main Success Scenario:

1. The Smart subscriber text money to 343.
2. The Smart subscriber accomplishes the application form.
3. The Smart subscriber presents valid ID to cashier and pays fee then gets receipt.
4. The Smart subscriber goes to customer service to give application form.
5. The Smart subscriber waits for the Smart Money Card to arrive or personally get it in 3 weeks.

Alternate Sequence:

A1: Smart subscriber doesn't have load.

From 0

1a. Smart subscriber reloads his/her Smart sim card.
2a. back to 1

A2: Smart subscriber doesn't have enough money.
From 2
3a. gets the exact amount of money needed.
4a. back to 3

Error Sequence:

E1: Smart Wireless Center is closed.
From 0
1a. wait until Smart Wireless Center is open
2a. back to 1

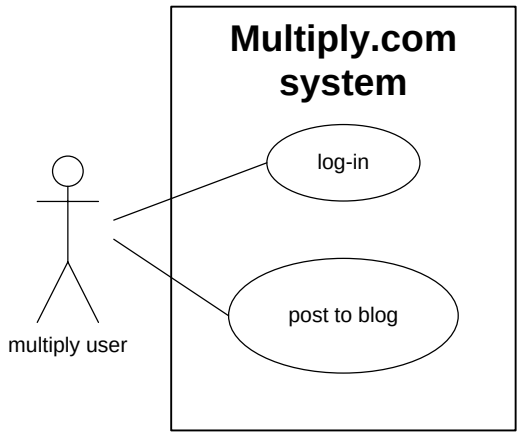
Post conditions:

1. Smart subscriber will have Smart Money Card.
2. Application form is less.
3. Smart subscriber has less money.

UI Requirements

1. Smart subscriber's sim card
2. Smart subscriber's money
3. Smart subscriber's valid ID

USE CASE # 7



Identification summary

Title: Multiply.com System
Summary: This use case allows a multiply user post blog.
Actors: multiply user
Creation Date: July 16, 2008
Version: 1.0
Date of Update: July 16, 2008
Person in charge: Lim, Marylyn Grace C.

Flow of Events

Precondition:

1. User must have an account in Multiply
2. User must have an internet connection.

Main Success Scenario:

1. The multiply user log-in to multiply.
2. The multiply user clicks on post.
3. The multiply user clicks on icon blog.
4. The multiply user composes blog entry.
5. The multiply user saves and publish blog.

Alternate Sequence:

A1: Incorrect username

From 1

2a. the multiply user types in the correct username.

3a. back to 2

A2: Incorrect password

From 1

2a. the multiply user types in the correct password.

3a. back to 2

Error Sequence:

E1: Server is down.

From 0

1a. wait until server is back to normal.

2a. back to 1

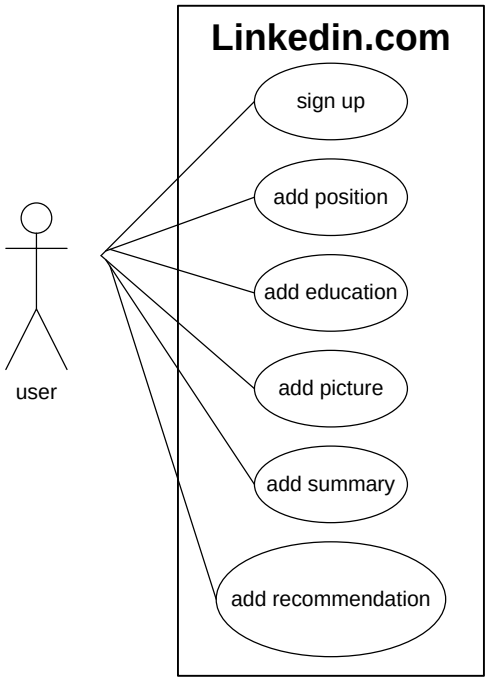
Post conditions:

1. The multiply user has a blog on his/her profile.

UI Requirements

1. User's multiply account.

USE CASE # 8



Identification summary

Title: Linkedin.com

Summary: This use case allows a user to make his/her profile 100% complete.

Actors: user

Creation Date: July 23, 2008

Version: 1.0

Person in charge: Lim, Marylyn Grace C.

Flow of Events

Precondition:

1. User must have an account in linkedin.
2. User must have an internet connection.

Main Success Scenario:

1. The user signs up to multiply.
2. The user fills up the information.
3. The user adds position.
4. The user adds education.
5. The user adds picture.
6. The user adds summary.
7. The user adds recommendation.

Alternate Sequence:

A1: The user hasn't confirmed his/her email address.

From 6

7a. the user confirms his/her email address.

8a. back to 7

A2: Another user is not in the contact of the user.

From 6

7a. the user invite the another user to his/her contacts.
8a. back to 7

A3: The picture isn't a valid file type.
From 4
5a. the user adds valid photo type.
6a. back to 5

Error Sequence:

E1: Server is down.
From 0
1a. wait until server is back to normal.
2a. back to 1

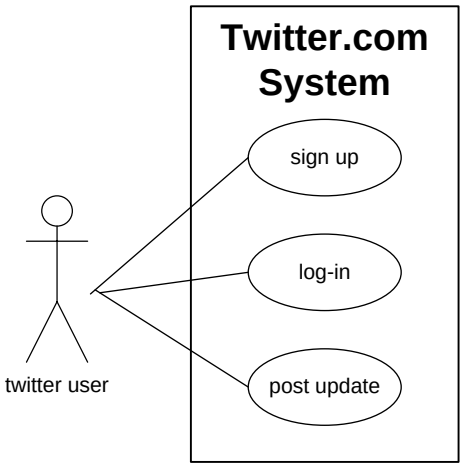
Post conditions:

1. The user has a 100% complete on his/her profile.

UI Requirements

1. User's linkedin account.

USE CASE # 9



Identification summary

Title: Twitter.com System
Summary: This use case allows a multiply user post updates
Actors: twitter user
Creation Date: July 29, 2008
Version: 1.0
Date of Update: July 29, 2008
Person in charge: Lim, Marylyn Grace C.

Flow of Events

Precondition:

1. User must have an account in Twitter.
2. User must have an internet connection.

Main Success Scenario:

1. The twitter user enters username.
2. The twitter user enters password.

3. The twitter user clicks on sign in.
4. The twitter user types in what they're doing.
5. The twitter user clicks on update.

Alternate Sequence:

A1: Incorrect username

From 1

2a. the twitter user types in the correct username.

3a. back to 3

A2: Incorrect password

From 2

2a. the twitter user types in the correct password.

3a. back to 3

Error Sequence:

E1: Server is down.

From 0

1a. wait until server is back to normal.

2a. back to 1

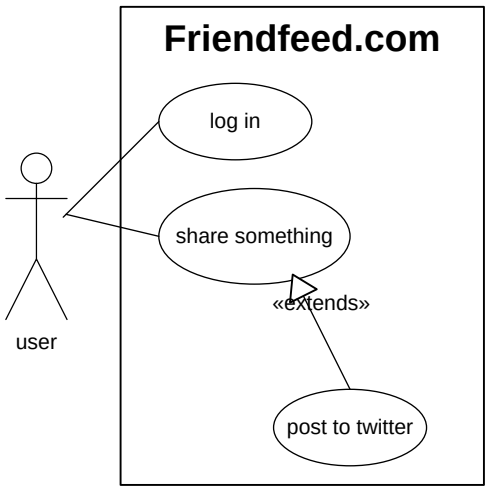
Post conditions:

1. The twitter user has already posted an update on his/her profile.

UI Requirements

1. User's twitter account.

USE CASE # 10



Identification summary

Title: Friendfeed.com System

Summary: This use case allows a user to post to twitter.

Actors: user

Creation Date: Aug. 6, 2008

Version: 1.0

Date of Update: Aug. 6, 2008

Person in charge: Lim, Marylyn Grace C.

Flow of Events

Precondition:

1. User must have an account in Friend feed.
2. User must have an account in Twitter.
3. User must have an internet connection.

Main Success Scenario:

1. The user accesses his/her friend feed account.
2. The user accesses the twitter website.
3. The user enters his/her twitter username.
4. The user enters his/her twitter password.
5. The user clicks sign in.
6. The user types in what they're doing.
7. The user clicks update.
8. The user clicks refresh on friend feed.

Alternate Sequence:

A1: Incorrect username

From 2

3a. the user types in the correct username.

4a. back to 4

A2: Incorrect password

From 3

4a. the twitter user types in the correct password.

5a. back to 5

Error Sequence:

E1: Server is down.

From 0

1a. wait until server is back to normal.

2a. back to 1

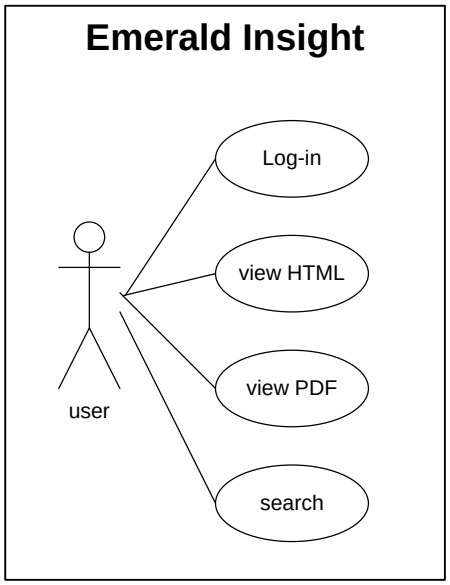
Post conditions:

1. The twitter user has already posted an update on his/her profile.

UI Requirements

1. User's twitter account.
2. User's friend feed account.

USE CASE # 11



Identification summary

Title: Viewing PDF file
Summary: This use case allows user to view PDF file.
Actors: user
Creation Date: Aug. 13, 2008
Version: 1.0
Date of Update: Aug. 13, 2008
Person in charge: Lim, Marylyn Grace C.

Flow of Events

Precondition:

1. User must have an internet connection.

Main Success Scenario:

1. The user finds the information he or she needed.
2. The user clicks on view PDF.

Alternate Sequence:

A1: Doesn't have the right to view the file.
From 1
2a. the user logs in or subscribe to it.
3a. back to 2

Error Sequence:

E1: From 0
1. Server is down.
- Use case fails.

E2: From 0
1. No internet connection
- Use case fails.

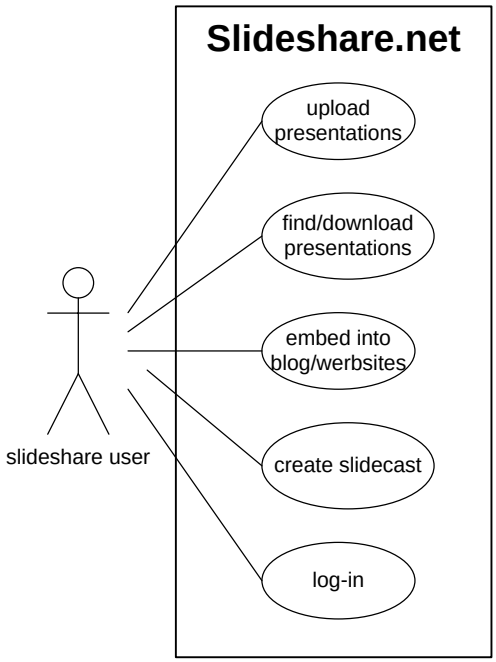
Post conditions:

1. The user was able to view the PDF file.

UI Requirements

1. Keywords for searching information.

USE CASE # 12



Identification summary

Title: Uploading Presentations
Summary: This use case allows a slideshare user to upload presentations
Actors: slideshare user
Creation Date: Aug. 18, 2008
Version: 1.0
Date of Update: Aug. 18, 2008
Person in charge: Lim, Marylyn Grace C.

Flow of Events

Precondition:

1. User must have an account in Slideshare.net.
2. User must have an internet connection.

Main Success Scenario:

1. The slideshare user clicks upload.
2. The slideshare user clicks browse and select files to be upload.
3. The slideshare user enters the details.
4. The slideshare user clicks publish.

Alternate Sequence:

A1: Invalid file format to upload

From 2

3a. the user picks on the correct file format to be uploaded

4a. back to 3

Error Sequence:

E1: Server is down.

From 0

1a. wait until server is back to normal.

2a. back to 1

Post conditions:

1. The multiply user has a blog on his/her profile.

UI Requirements

1. User’s slideshare account.
2. User’s document to be uploaded.

A Systems Analysis Study on the
Creating of account on Express Link
Of Bank of the Philippine Islands

Presented to the
Information System Program
School of Management and Information Technology
De La Salle – College of Saint Benilde

In Partial fulfillment of the
Requirements of the subject
Systems Analysis

Submitted By:
Gonzaga, Andrew
Lim, Marylyn Grace
Santiago, Ian Jenesis
Dorado, Jan Aldrich
SysAnal O0A
August 15, 2008

Submitted To:
Mr. Paul Pajo

CHAPTER 1

Bank of the Philippine Islands

España corner Moret St., Sampaloc, Manila

Company Background

Statement of the purpose/ BPI Mission:

WE BELIEVE in the central role that private enterprise plays in economic development.

WE BELIEVE that our corporate mission is to be the leading private financial institution in the Philippines in terms of professional competence, service quality, responsible corporate citizenry, and overall growth and stability; and to be an established ASEAN financial institution with a creditable worldwide outreach.

WE BELIEVE that we have a responsibility to manage the business for the maximum benefit of our customers while adopting the highest standard of integrity; to offer the widest possible range of financial services that is responsive to their needs; and to adopt an objective attitude towards change and innovation, ever mindful of improving service quality and operating efficiency.

WE BELIEVE that we have a responsibility to develop the potential of our employees to the fullest by providing an environment conducive to their personal and professional growth; and to foster a value system held in common throughout the institution in order that we may all share a coherent sense of purpose and direction.

WE BELIEVE that we have a responsibility to attain, over time and within exacting standards of prudent management, the highest possible return on the investment of our shareholders

History of BPI

Philippine banking has a long and colorful history. It began in 1828 when, as the Philippines reaped the benefits of increased trade, King Ferdinand VII of Spain issued a decree mandating the establishment of a public bank in the Philippines.

The first managers of the Bank were Jose Maria Tuason and Fernando Aguirre, who each took turns serving as managing director every year. While the members of the Bank's highest policy-making board were essentially civil and ecclesiastical officials, there was also a businessman whom the Spanish Crown named to represent the business community of Manila. The man was Antonio de Ayala of the prominent Casa Roxas, precursor of Ayala y Cia, which is now Ayala Corporation.

The royal decree that confirmed the creation of El Banco Español Filipino de Isabel 2 also gave the Bank the exclusive privilege to issue paper money, which antedated the currency-issuing authority of the post-war Central Bank of the Philippines by about a hundred years (The present central bank, the official issuer of Philippine currency, started operations only in 1949). The original bank notes were collectively called pesos fuertes (PF), Spanish for "strong pesos."

The first bank notes (or paper money) in the Philippines had the issue date May 1, 1852 and could be redeemed in Mexican coins in gold or silver. Apart from carrying the name of the Bank as issuer of the currency, the bank notes also bore the portrait of the woman for whom the bank was named –- Queen Isabella II.

Coincidentally, the first transaction of the Bank was a lending transaction recorded on May 1, 1852, in which the Bank discounted a promissory note from a Chinese client. Three days later, the Bank recorded its first deposit from its first depositor.

Today, BPI has maintained a leadership position in consumer banking, trust banking, and asset management, corporate banking/corporate finance, and bancassurance. With over 700 branches and around 1,100 automated teller machines, BPI boasts of having the largest combined network of branches/kiosk units and ATMs, servicing some three million depositors.

For years, international publications and rating agencies have given annual awards to BPI as one of the best banks in the region. Among these are Asiamoney, BusinessWeek, Euromoney, Far Eastern Economic Review, Finance Asia, Global Finance, The Asian Banker, The Asset, and The Banker.

Products and Services

Product

- Deposit
- Loans
- ATM/ debit cards
- Credit Cards
- Prepaid Cards
- Insurance
- Remittance
- Global Transactions
- Asset Management
- Foreign Exchange
- Leasing

Services

- Mobile Banking
- Phone Banking
- Internet Bankint
- ATM Banking
- Cash Acceptance Machine
- Bill Payment Partners
- Prepaid Cellphone Reloading

Statement of the Problem

- Creating an account is a very important task in a bank, it only takes a short time, about 20 to 30 minutes tops, but you still need to consider the time and the cost of transportation it takes to travel to a bank twice if your schedule for the interview is on a different day.
    1. The client does not have the time to make a bank account.

2. Some clients have to return back in a given date by the bank to conduct the interview.

- The cause of this problem is when a client applies in a bank to create an account, and the banks allows it, then the client will ask if what is needed to create an account and the bank tells the client the requirement, the applicant would return on another day while brining the requirements and its either the applicant would have the interview on that day or the applicant would return another day for the interview, as you can see it takes too much time and effort on the applicants side.

- We chose this process because it is one of the most important system that a bank has, if the creating of account would make the clients or applicant irritated then there would be less applicant, and if there is less applicant then the bank would have a grow slowly. So if we try to make the process faster, then it would have an effect in the bank's growth.

Objectives of the System

- By the problems that they've been encountering on the daily hours of banking, we came up with a proposal which we call it mobile registration with an idea to help the system to be fast in productivity, less complication and more reliable sources. Now our goals is to satisfy the customer's needs as a result of our proposed systems we want to make the process of creating an account in BPI faster, easier and more convenient.

- The proposed system is an express mobile registration for opening a new account service, meaning you can use your mobile phones by simply texting to a certain number of the bank, and then you will receive information regarding the requirements and you would be able to pick available schedule for you interview.

- Having this new system will allow faster processing and to help improve the overall performance in creating an account. In our proposal system, we aim to improve more to these following existing problems.

    - Improve the process time.
    - Make it more convenient for the applicants.
    - If the applicant applies via text, the bank will be notified and will be more prepared.
    - To improve the quality of service in creating an account with in BPI bank.

- We will be able to address that problem by adding and changing some things inside their process.
    - Adding a cell phone to receive text.
    - Allowing answering of personal important information using cellular phone.

Significance of the Study

One of the most important systems a bank has is the New Account System. It is where the banking transaction starts. So the bank has an important role in the financial system of a country. Basically the banking system has many to offer when you do open a new account.

So based on the problem that we've discovered, we have seen that these problems will affect the numbers of applicants for creating a new account in the bank, and in improving the system in creating new account, would result to the applicants would have more conveniently experience in creating an account, and that will also affect on how a bank grows.

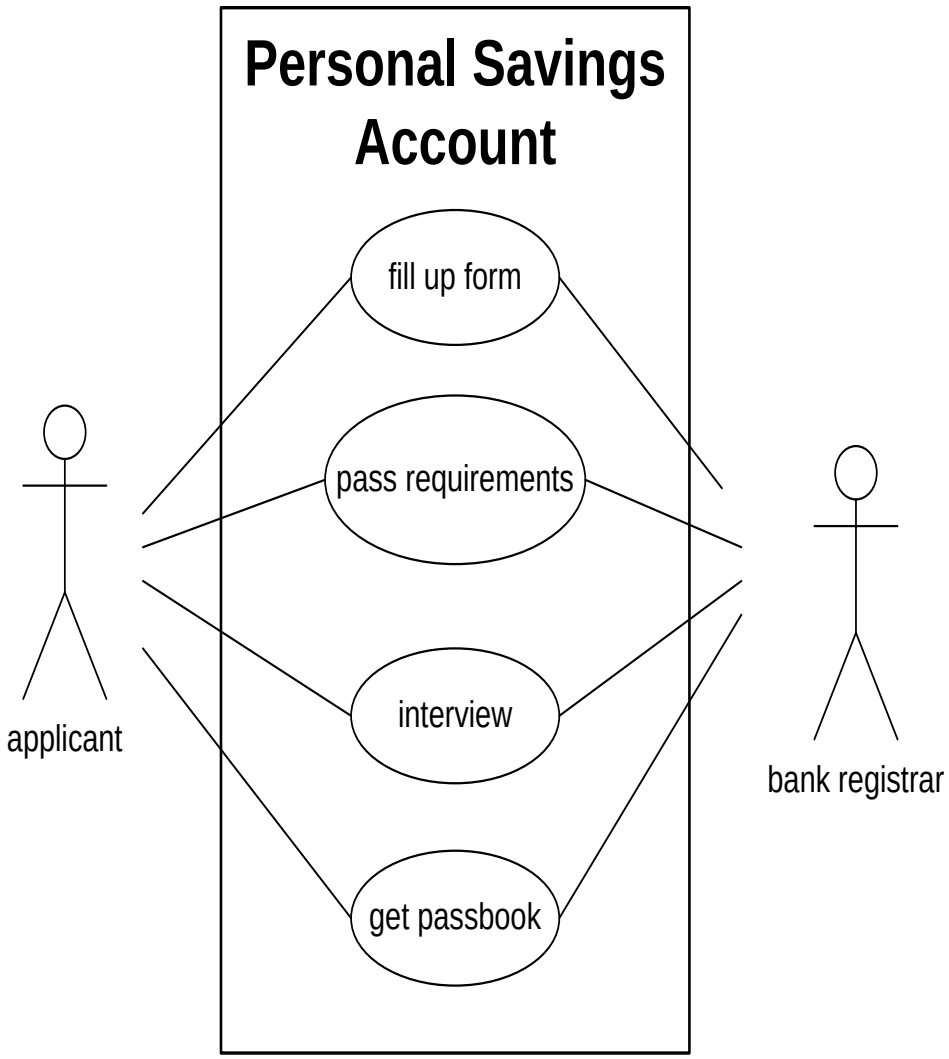
The importance of our study in this system is to make the system of creating an account convenient for the applicants and to have other option in creating an account instead of just going to the bank yourself.

Scope and Limitation

- In our study we focused on adding a new feature in create an account in BPI bank, and that feature is creating an account via mobile phone. Our scope would be the executing system of creating an account which is requiring the applicant to go to the bank. We would not tackle the process of how an account is being saved into the data base of the main server.

CHAPTER 2

Use Case Diagram of the Existing System



Use case 1 - Fill up form

Identification Summary

Title: Filling up customer information sheet

Summary: This use case allows an applicant to fill up customer information sheet

Actors: applicant

Creation Date: Aug.13, 2008

Version: 1.0

Date of Update: Aug. 14, 2008

Person in charge: Lim, Marylyn Grace C.

Flow of Events

Precondition:

1. The bank must have customer information sheet.

Main Success Scenario:

1. The applicant gets customer information sheet.
2. The applicant fills up customer information sheet.
3. The applicant passes the customer information sheet.

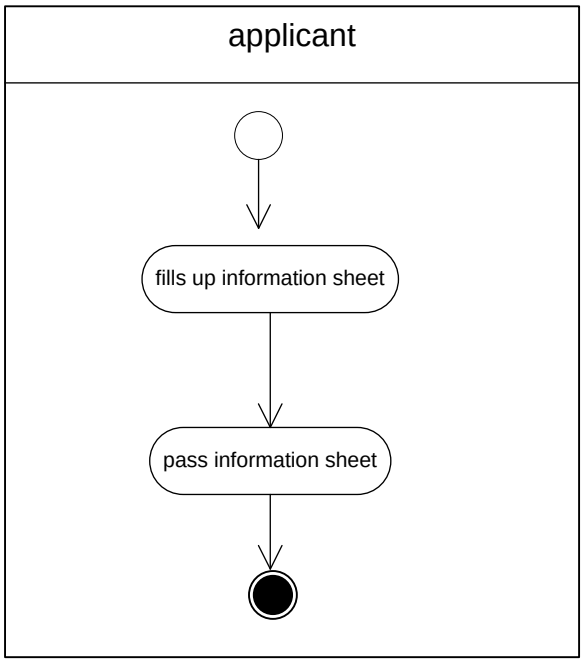
Post conditions:

1. Customer information sheet became lesser.
2. The applicant was able to pass his/her customer information sheet.

Non-functional Requirements:

Response Time: The entire transaction can be done in 15 minutes.

Activity 1- **Fill up form**



Use case 2 - Pass Requirements

Identification Summary

Title: Passing requirements

Summary: This use case allows an applicant to pass their requirements.

Actors: applicant, bank registrar

Creation Date: Aug.13, 2008

Version: 1.0

Date of Update: Aug. 14, 2008

Person in charge: Lim, Marylyn Grace C.

Flow of Events

Precondition:

1. The applicant must have any 2 valid IDs.
2. The applicant must have a 2x2 picture.
3. The applicant must have a proof of billing address.

Main Success Scenario:

1. The applicant passes the requirements.
2. The bank registrar checks the requirements.

Alternate Sequence:

A1: incomplete requirements

From 0

1a. completes needed requirements

2a. back to 1

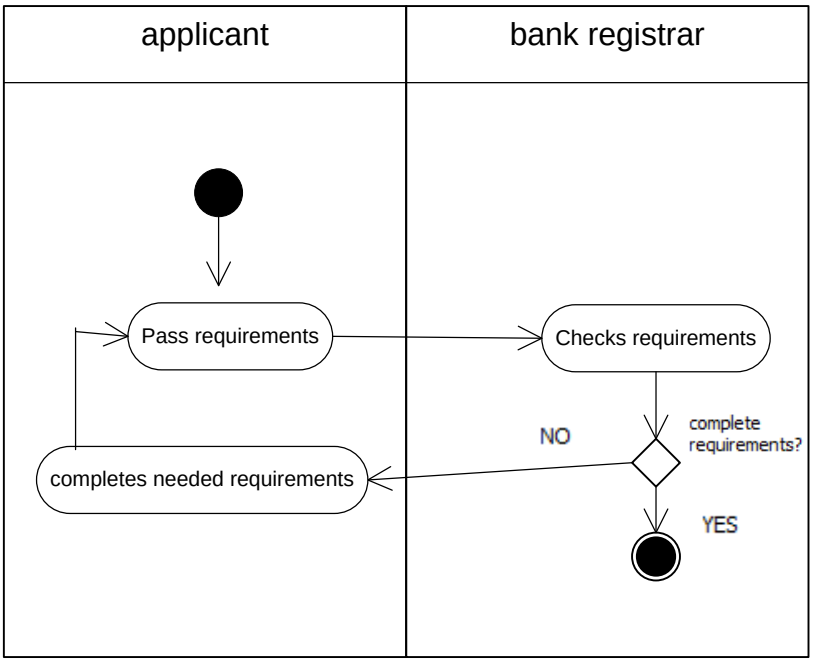
Post conditions:

1. The applicant was able to pass the requirements.

Non-functional Requirements:

Response Time: The entire transaction can be done in 5 minutes.

Activity 2**- Pass Requirements**



Use case 3 - Interview

Identification Summary

Title: Interviewing applicant

Summary: This use case allows the bank registrar to conduct an interview.

Actors: applicant, bank registrar

Creation Date: Aug.13, 2008

Version: 1.0

Date of Update: Aug. 14, 2008

Person in charge: Dorado, Jan Aldrich J.

Flow of Events

Precondition:

1. The applicant has passed complete requirements.

Main Success Scenario:

1. The bank registrar asks questions.
2. The applicant answers the bank registrar's questions.

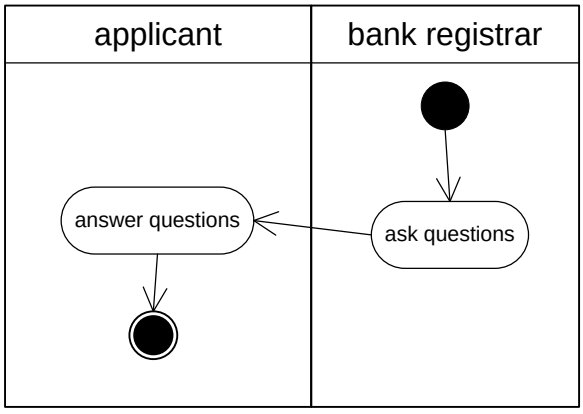
Post conditions:

1. The bank registrar was able to conduct the interview.

Non-functional Requirements:

Response Time: The entire transaction can be done in 15 minutes.

Activity 3- **Interview**



Use case 4 - Get passbook

Identification Summary

Title: Getting Passbook

Summary: This use case allows an applicant to get their passbook.

Actors: applicant, bank registrar

Creation Date: Aug.13, 2008

Version: 1.0

Date of Update: Aug. 14, 2008

Person in charge: Lim, Marylyn Grace C.

Flow of Events

Precondition:

1. The applicant has finished the interview.

Main Success Scenario:

1. The bank registrar prints the passbook.
2. The bank registrar inputs data.
3. The bank registrar gives the passbook to applicant.
4. The applicant receives the passbook.

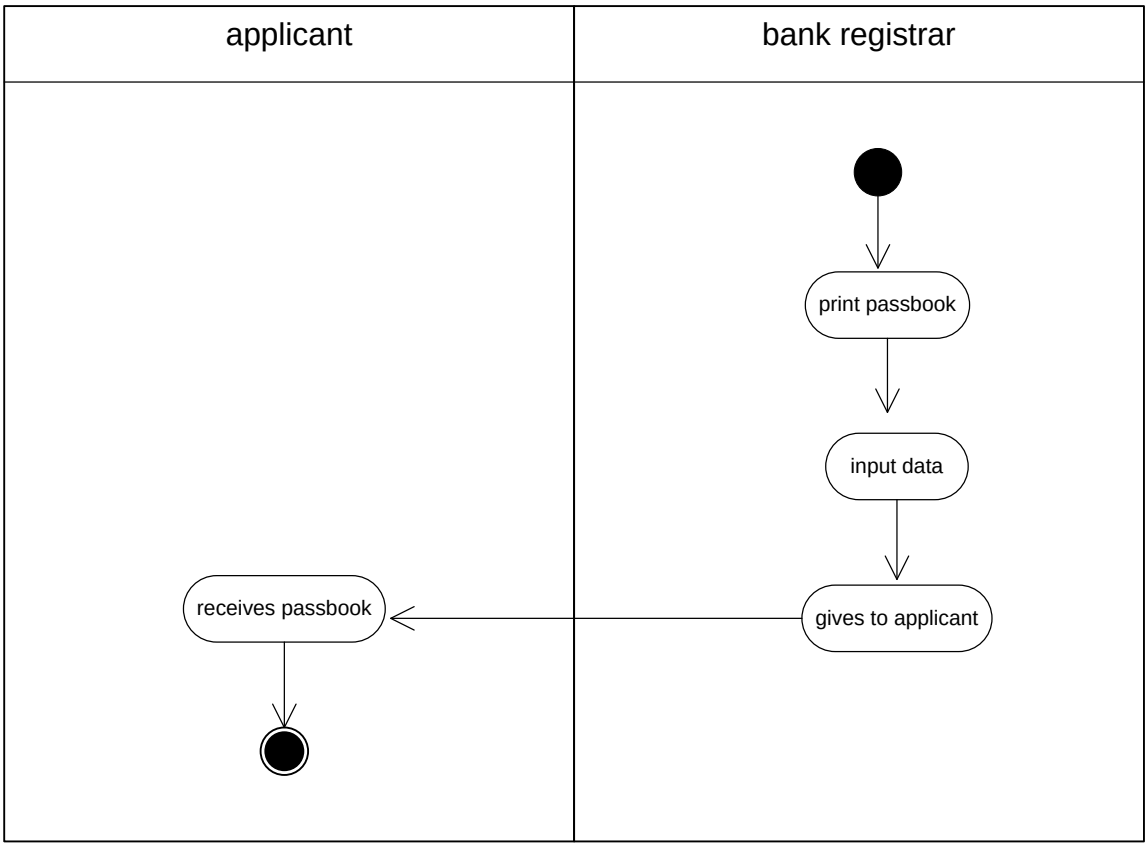
Post conditions:

1. The applicant has successfully created a savings account.

Non-functional Requirements:

Response Time: The entire transaction can be done in 10 minutes.

Activity 4**- Get passbook**



Processing Time vs. Cycle Time

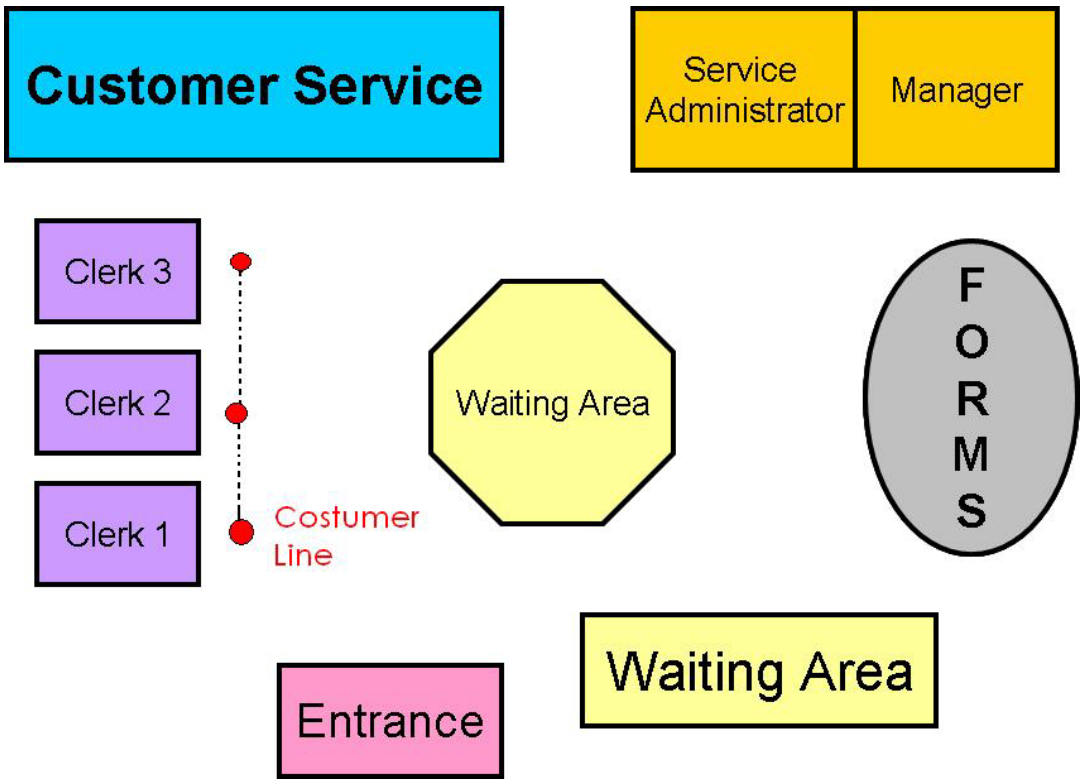
Existing System

Activities	Processing Time	Cycle Time
1. fill up form	10 minutes	10 minutes
2. pass requirements	3 minutes.	5 minutes
3. interview	10 minutes	15 minutes
4. get passbook	7 minutes	10 minutes
Total:	30 minutes	40 minutes

Proposed System

Activities	Processing Time	Cycle Time
1. text personal information	3 minutes	5 minutes
2. interview	10 minutes	15 minutes
3. get passbook	7 minutes	10 minutes
Total:	20 minutes	30 minutes

Geographical Flowchart



CHAPTER 3

Table of Recommendations

Problems Observed	Recommended Changes	Affected Activities
1. Clients do not have time to make a bank account.	Send the list of requirements to the applicant via text.	Passing of requirements(client)
2. Some clients have to return back in a given date by the bank to conduct the interview.	The process of interview should be after the requirements are passed.	Passing of requirements.(client and customer service) Interview. (Interviewer)

Streamlining

We used 2 tools of the 12 streamlining tools and those tools are:

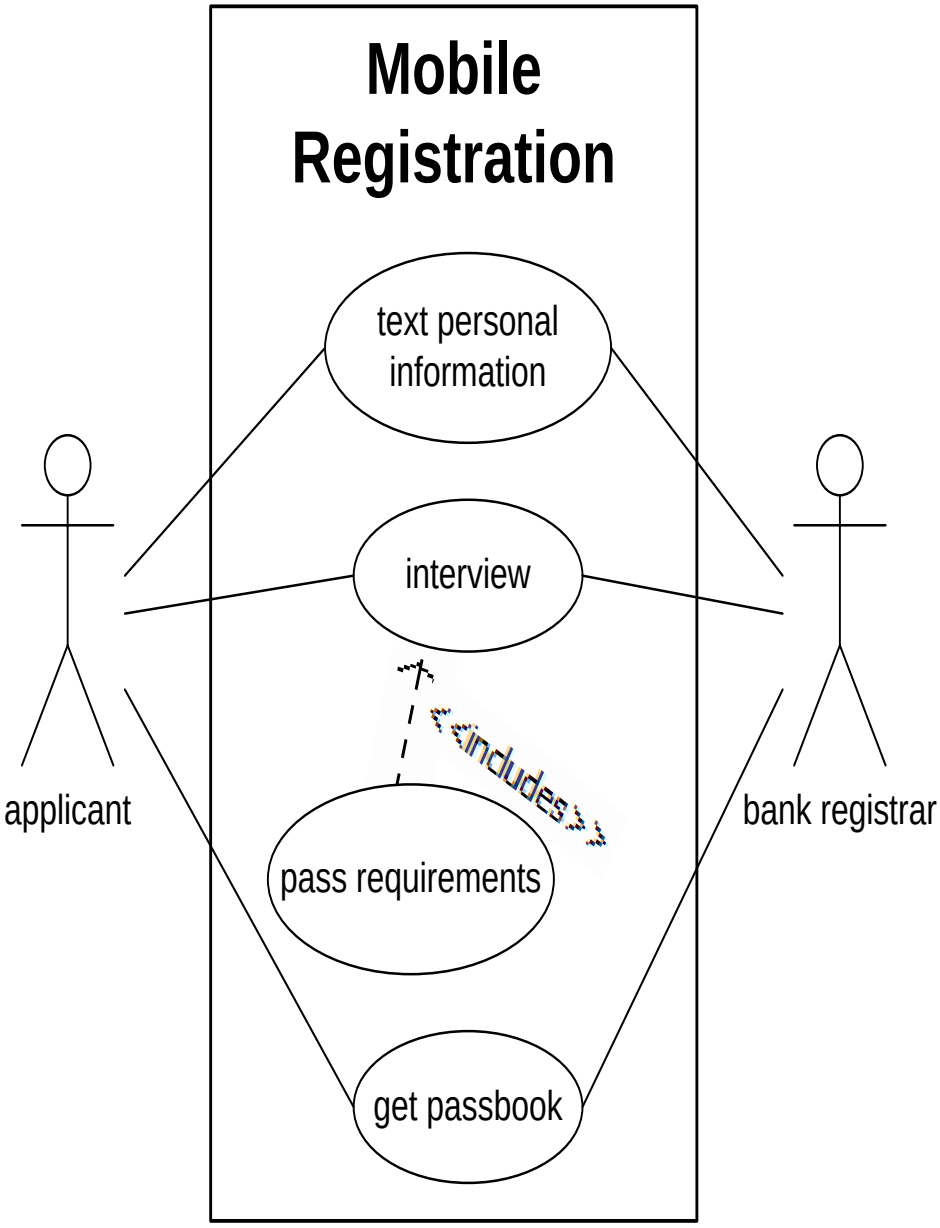
1. Process cycle-time reduction:

With our proposed system, it would make it more convenient for the client to make an account using the mobile, instead of going to the bank twice, because of the different day of the interview.

2. Upgrading:

We upgraded the creating an account system by adding a new feature and that is you can set your interview within the bank available schedule also, with this you don't need to return to the bank if your interview is scheduled on a different day.

Use Case Diagram of the Proposed System



Use case 1 - Text personal information

Identification Summary

Title: Texting Personal Information

Summary: This use case allows an applicant to register thru mobile.

Actors: applicant, bank registrar

Creation Date: Aug.13, 2008

Version: 2.0

Date of Update: Aug. 14, 2008

Person in charge: Lim, Marylyn Grace C.

Flow of Events

Precondition:

1. The applicant should have load.

Main Success Scenario:

1. The applicant texts personal information.
2. The bank registrar receives text.
3. The bank registrar sends available schedule.
4. The applicant picks desired schedule and sends it to bank registrar.
5. The bank registrar receives the schedule.
6. The bank registrar sends the confirmation number and requirements.
7. The applicant receives the confirmation message.

Error Sequence:

E1: from 1

-Applicant doesn't have load.

-Use case fails

Post conditions:

1. The applicant was able to get the confirmation number.
2. The applicant was able to register thru mobile.

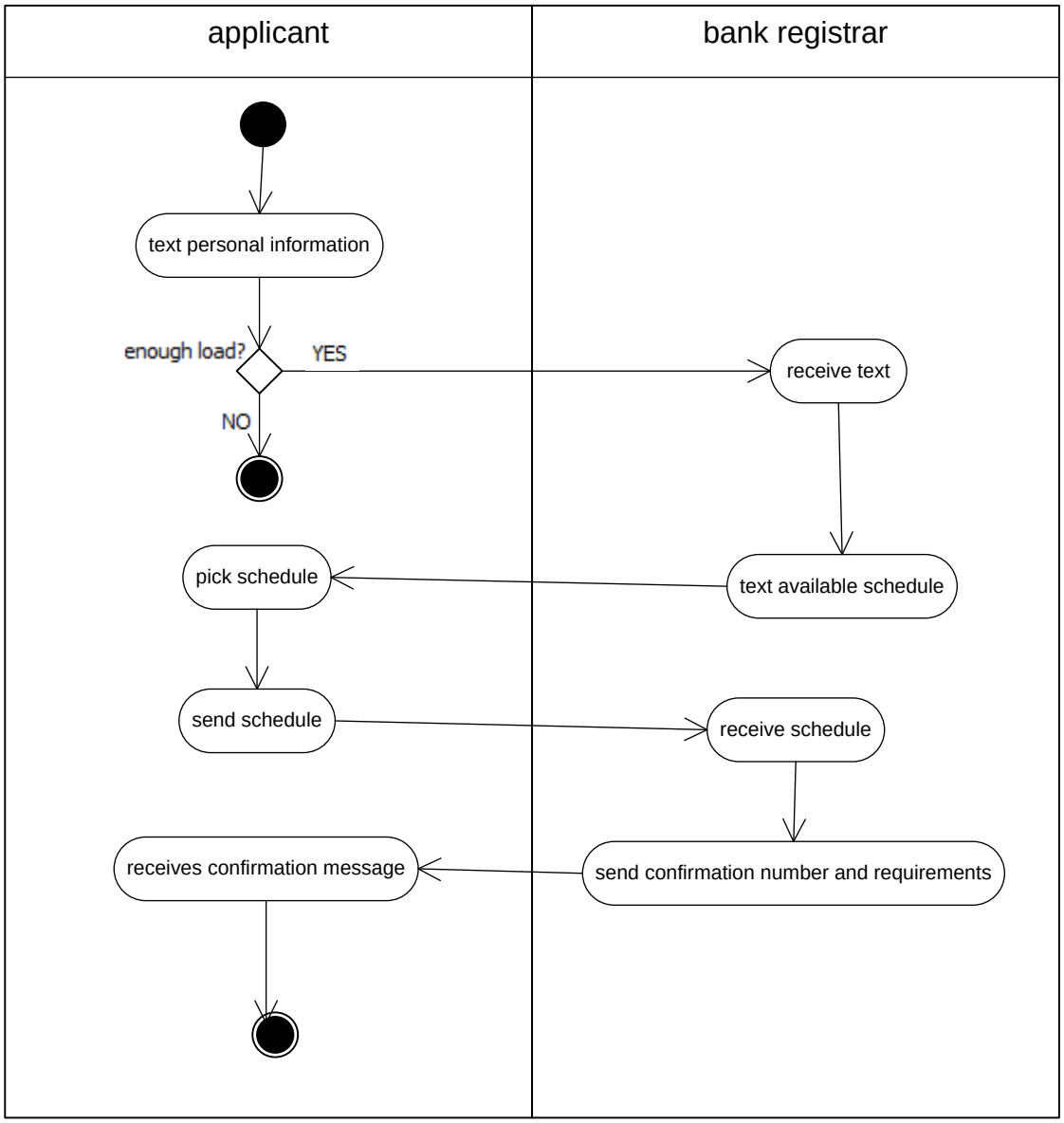
U.I. Requirements:

1. Input: personal information, schedule
2. Output: confirmation number

Non-functional Requirements:

Response Time: The entire transaction can be done in 5 minutes.

Activity 1- **Text personal information**



Use case 2 - Interview

Identification Summary

Title: Giving requirements and interview

Summary: This use case allows an applicant to pass requirements and have been interviewed.

Actors: applicant, bank registrar

Creation Date: Aug.13, 2008

Version: 2.0

Date of Update: Aug. 14, 2008

Person in charge: Lim, Marylyn Grace C.

Flow of Events

Precondition:

1. The applicant must have any 2 valid IDs.
2. The applicant must have a 2x2 picture.
3. The applicant must have a proof of billing address.
4. The applicant must have a confirmation numbers.

Main Success Scenario:

1. The applicant goes to BPI.
2. The applicant gives requirement and confirmation number to bank registrar.
3. The bank registrar checks requirements and confirmation number.
4. The bank registrar starts interviewing the applicant.

Alternate Sequence:

A1: incomplete requirements

From 3

4a. Bank registrar gives the incomplete requirements back to the applicant

5a. Applicant completes the needed requirements

6a. go back to 2

Error Sequence:

E1: bank is closed.

-use case fails

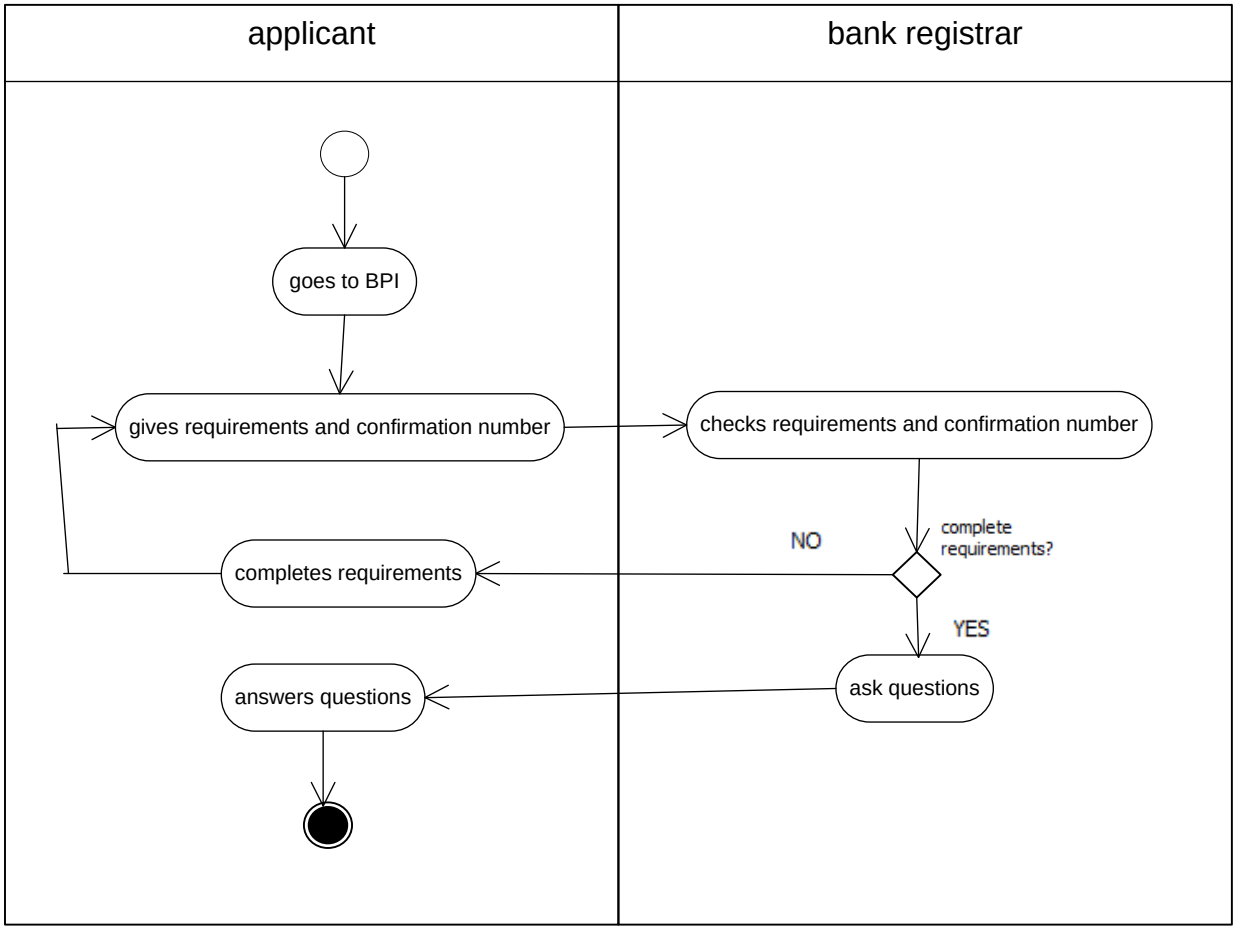
Post conditions:

1. The applicant was able to pass the requirements.
2. The bank registrar was able to conduct an interview.

Non-functional Requirements:

Response Time: The entire transaction can be done in 15 minutes.

Activity 2**- Interview**



Use case 3 - Get passbook

Identification Summary

Title: Getting Passbook

Summary: This use case allows an applicant to get their passbook.

Actors: applicant, bank registrar

Creation Date: Aug.13, 2008

Version: 2.0

Date of Update: Aug. 14, 2008

Person in charge: Lim, Marylyn Grace C.

Flow of Events

Precondition:

1. Finished the interview.

Main Success Scenario:

1. The bank registrar prints the passbook.
2. The bank registrar inputs data.
3. The bank registrar gives the passbook to applicant.
4. The applicant receives the passbook.

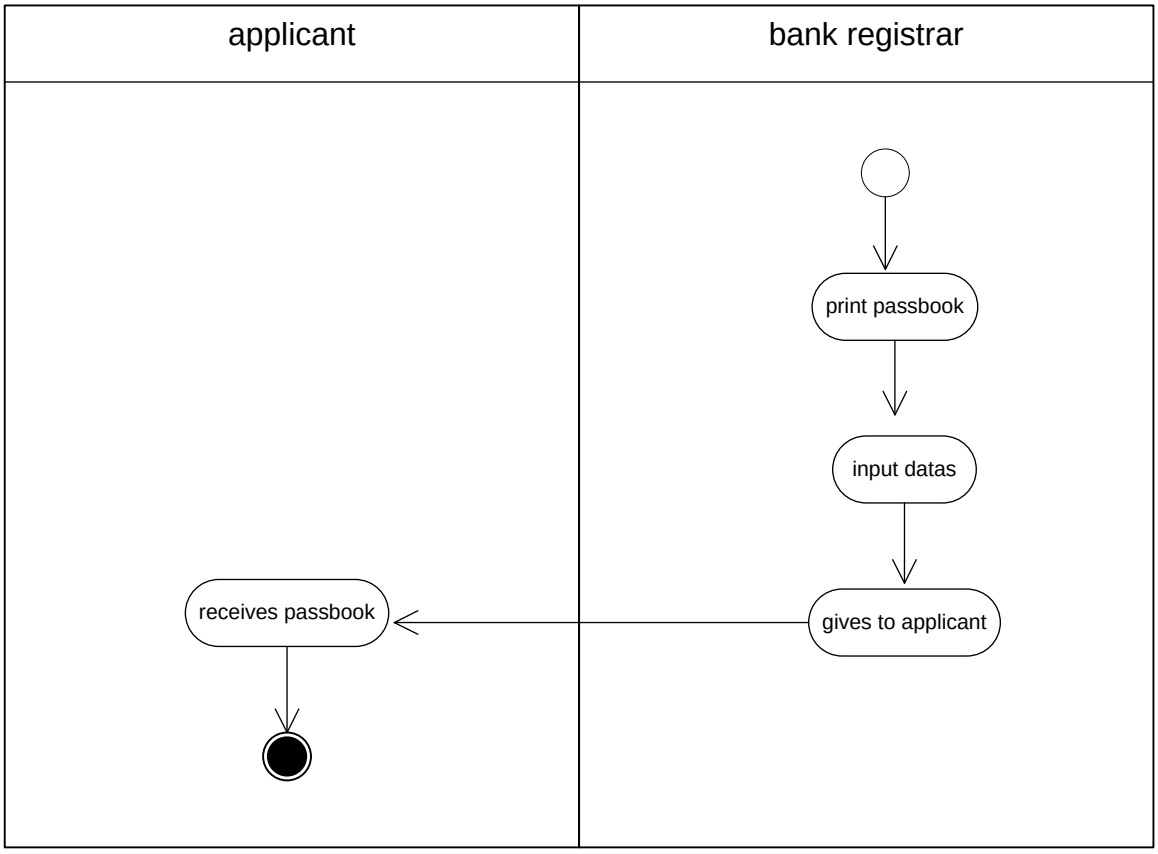
Post conditions:

1. The applicant has successfully created a savings account.

Non-functional Requirements:

Response Time: The entire transaction can be done in 10 minutes.

Activity 3- **Get passbook**



REFERENCE

http://www.amazon.com/System-Analysis-Design-Active-Approach/dp/0536633452/

http://www.twitter.com

http://www. multiply.com

http://www.slideshare.net

http://www.emeraldinsight.com/Insight/menuNavigation.do;jsessionid=09F114E643442815B75C6CE78B4AF7DF?hdAction=InsightHome

http://www.friendfeed.com

http://www.linkedin.com