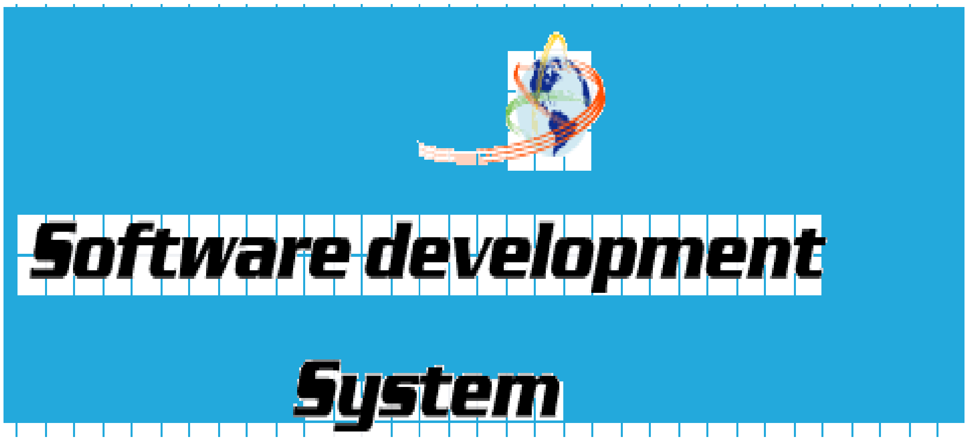
TEESSIDE UNIVERSITY
SCHOOL OF COMPUTING

System Development methods

Kiril Anastasov - L1087591



Module leader:
Mansha Nawaz

Table of Contents

Table of Figures

Abstract

The aim of this report is to create a software development system that creates software for customers. The customers will be able to give their requirements to the system and then the system will design and develop the software. The customers can contact the company through website or telephone number. Prices are often given for existing projects to give an idea how much a standard projects costs. The design will be covered later by the data dictionaries and web design.

There are going to be a lot of customers so a lot of information is going to come from the customers and a lot of time will be spend when getting the requirements for the system.

The research that is intended to be done will cover the methodologies available in order to complete this documentation.

Acknowledgements

I am grateful for the help and support of my spiritual and mental leader Mr Mansha Nawaz and Thomas Philpot. Without the on-going support they gave me, it would not have been possible to complete the report. They were able to get in touch despite the time and place, in order to help for achieving all goals. They made the effort to personally get to know me and my project and give feedback where it was needed.

1. Project introduction

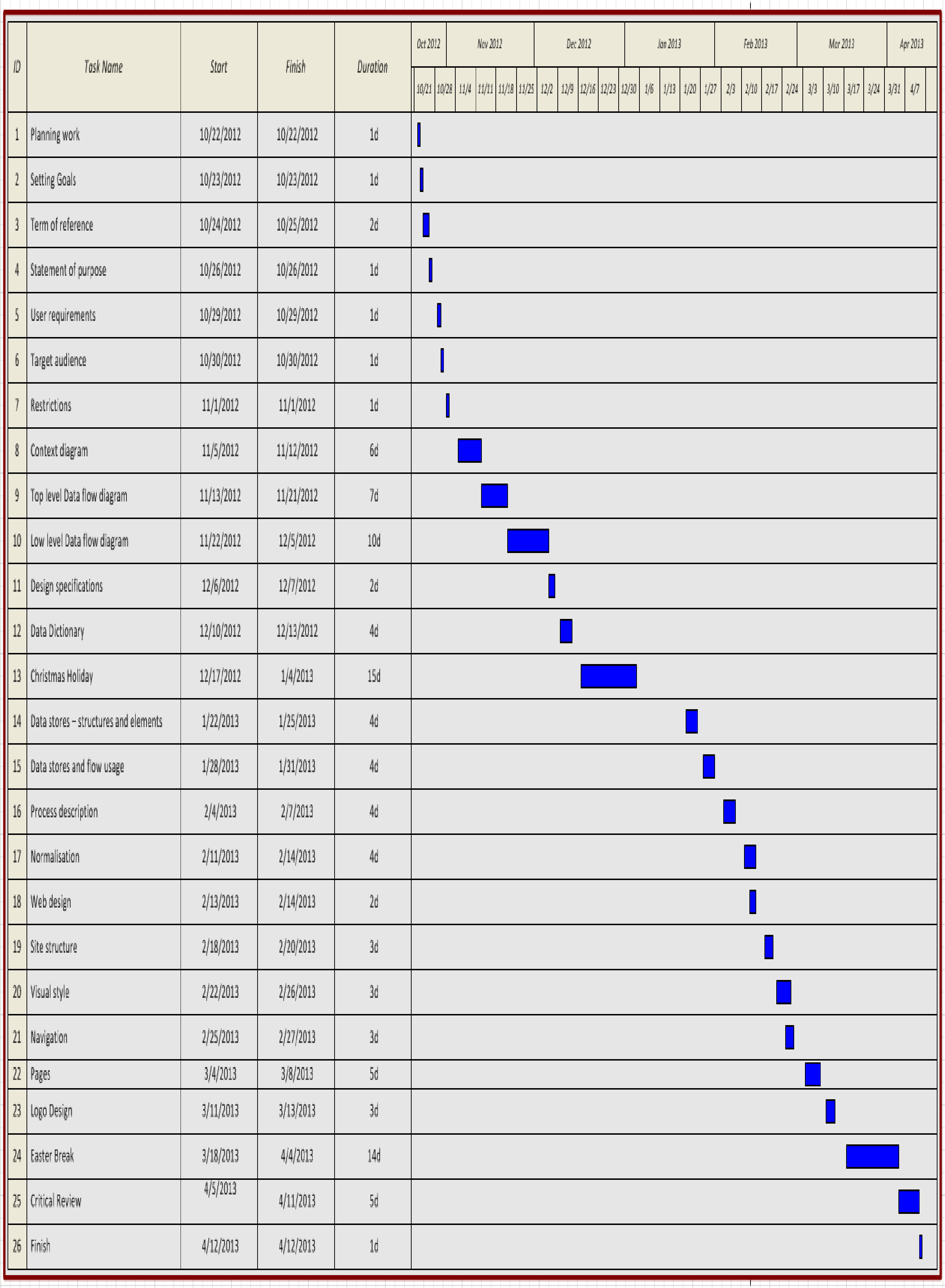
As this project is not going to be developed for or around a company, the ideas and functions of the system come straight from me. In other situations or other projects, a developer will have a client they would be able to withdraw the information and requirements from, either by face to face interviews or questionnaires. As this advantage is not present and developing the idea from scratch, more research will need to be done in to the field and try and take advice from these existing companies. This will hopefully give the system the edge to enhance on these existing companies (which my requirements will come from) and add my ideas to them.

1.1 Project Proposal and Scope

For this project the aim is to develop a software development system. The system will be developed with a website so the customers can access it fast and easily. The system will allow customers to contact the company and discuss their needs for software. The company will provide a wide range of services like database development, web development and design, mobile development, desktop development and software consulting. On the website there is going to be a section with a previous customer's reviews and opinions after they have used the company's services. The company hopes to attract new customers through their website, the recommendation of the current customers and advertising on the social networks like Facebook, twitter and LinkedIn.

1.2 Planning and scheduling work

The system has until April 17th to be complete the Analysis and Design documentation for this project, so the work could be spread out evenly so it is manageable. Depending on the classes times we have in the university and holidays that are available this Gantt chart will display how the work will be spread out. The main holidays will be as follows: Winter break: 17th December – 4th January, Reading week: 25th February – 3th March and Spring break: 17th March – 8nd April. The gannt chart is accompanied by a task list to the left. The project should be finished just after the spring break holidays, giving enough time (2 weeks) to read over and change any sections that needs too.



ID	Task Name	Start	Finish	Duration
1	Planning work	10/22/2012	10/22/2012	1d
2	Setting Goals	10/23/2012	10/23/2012	1d
3	Term of reference	10/24/2012	10/25/2012	2d
4	Statement of purpose	10/26/2012	10/26/2012	1d
5	User requirements	10/29/2012	10/29/2012	1d
6	Target audience	10/30/2012	10/30/2012	1d
7	Restrictions	11/1/2012	11/1/2012	1d
8	Context diagram	11/5/2012	11/12/2012	6d
9	Top level Data flow diagram	11/13/2012	11/21/2012	7d
10	Low level Data flow diagram	11/22/2012	12/5/2012	10d
11	Design specifications	12/6/2012	12/7/2012	2d
12	Data Dictionary	12/10/2012	12/13/2012	4d
13	Christmas Holiday	12/17/2012	1/4/2013	15d
14	Data stores – structures and elements	1/22/2013	1/25/2013	4d
15	Data stores and flow usage	1/28/2013	1/31/2013	4d
16	Process description	2/4/2013	2/7/2013	4d
17	Normalisation	2/11/2013	2/14/2013	4d
18	Web design	2/13/2013	2/14/2013	2d
19	Site structure	2/18/2013	2/20/2013	3d
20	Visual style	2/22/2013	2/26/2013	3d
21	Navigation	2/25/2013	2/27/2013	3d
22	Pages	3/4/2013	3/8/2013	5d
23	Logo Design	3/11/2013	3/13/2013	3d
24	Easter Break	3/18/2013	4/4/2013	14d
25	Critical Review	4/5/2013	4/11/2013	5d
26	Finish	4/12/2013	4/12/2013	1d

1.3 Setting goals

The goals for this system are to analyze and design the best way to create the software development system, looking at what the system has to do, who it will be for, how the system will look, what documentation and diagrams need to go in to the system for it to work properly. The end goal is to have a completed document so that an outside user can read it, and know what the system is about and how to make it. This has been chosen to be the main goal as there are time, knowledge and skills restrictions. It is strongly desirable the documentation to be readable to developers like me and others so they could go on and implement the proposed system when the time is right. If that is possible the document will have achieved its main purpose and service.

1.4 Methodologies available

As there are many different methodologies available, the chosen are the six most popular to choose from, the waterfall method, prototyping, incremental development, spiral development and rapid application development. Each one has strengths and weaknesses but one needs to be chosen. Each of them will be reviewed and a decision will be made based on which one is the most appropriate.

1.4.1 Waterfall development

Rising (2009) provided information about the waterfall methodology is a sequential development approaching which developing the application is done in steps. The project is split up in to 6 sections. The sections are Analysis, Design, implementation (build), testing, integration and maintenance. The method allows you to have some overlap when crossing over stages, as they will integrate together.

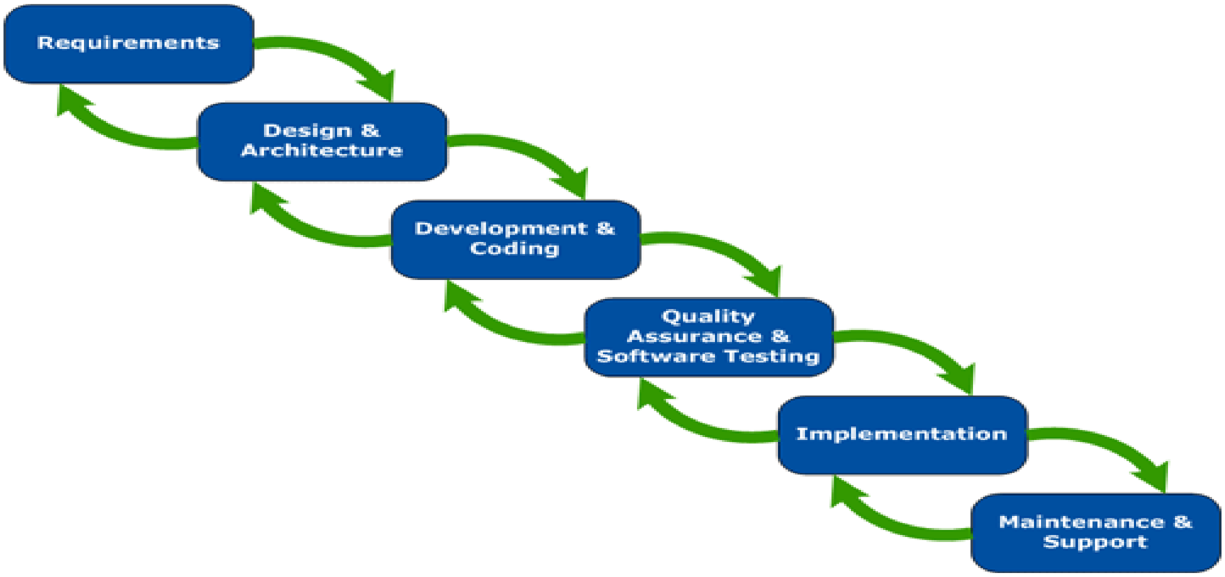


Figure 1: Waterfall development process (Rising 2009)

Advantages:

- ✓ Time spent early in the software production cycle can lead to greater economy at later stages.
- ✓ If a program design turns out to be impossible to implement, it is easier to fix the design at the design stage than to realize months later, when program components are being integrated, that all the work done so far has to be scrapped because of a broken design.
- ✓ Time spent early on making sure requirements and design are correct saves much time and effort later. Thus, the thinking of those who follow the waterfall process goes, make sure each phase is 100% complete and absolutely correct before proceeding to the next phase.
- ✓ Emphasizing on documentation (such as requirements documents and design documents) as well as source code. If members leave before the project is completed, it will be easier for the project to recover from a loss.
- ✓ Some waterfall proponents prefer the waterfall model for its simple approach and argue that it is more disciplined. The waterfall model provides a structured approach; the model itself progresses linearly through discrete, easily understandable and explainable phases and thus is easy to understand.

Disadvantages

- ✓ Some argue that the waterfall model is a bad idea in practice—believing it impossible for any non-trivial project to finish a phase of a software product's lifecycle.
- ✓ Designers may not be aware of future implementation difficulties when writing a design for an unimplemented software product.
- ✓ Many of the system's details only become known as implementation of the system progresses. Some of the things that are learnt invalidate the design and backtracking must occur.

1.4.2 Prototyping

According to MBA knowledge Base (2010) software prototyping refers to the activity of creating prototypes of software applications, incomplete versions of the software program being develop. A prototype typically simulates only a few aspects of, and may be completely different from, the final product.

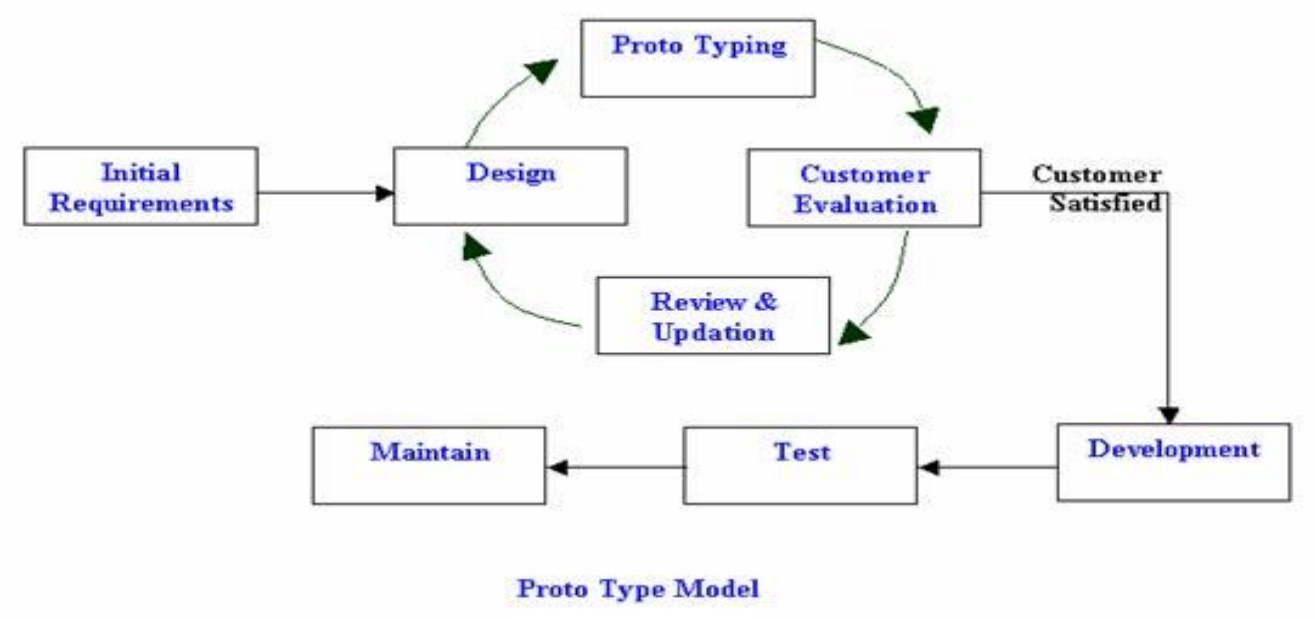


Figure 2: Proto Type Model (MBA knowledge Base, 2010)

Advantages:

- ✓ Reduced time and costs: Prototyping can improve the quality of requirements and specifications provided to developers. Because changes cost exponentially more to implement as they are detected later in development, the early determination of *what the user really wants* can result in faster and less expensive software.
- ✓ Improved and increased user involvement: Prototyping requires user involvement and allows them to see and interact with a prototype allowing them to provide better and more complete feedback and specifications. The presence of the prototype being examined by the user prevents many misunderstandings and miscommunications that occur when each side believe the other understands what they said.

Disadvantages:

- ✓ Insufficient analysis: The focus on a limited prototype can distract developers from properly analyzing the complete project. This can lead to overlooking better solutions, preparation of incomplete specifications or the conversion of limited prototypes into poorly engineered final projects that are hard to maintain.

- ✓ User confusion of prototype and finished system: Users can begin to think that a prototype, intended to be thrown away, is actually a final system that merely needs to be finished or polished. This can lead them to expect the prototype to accurately model the performance of the final system when this is not the intent of the developers.

- ✓ Developer misunderstanding of user objectives: Developers may assume that users share their objectives (e.g. to deliver core functionality on time and within budget), without understanding wider commercial issues.

- ✓ Developer attachments to prototype: Developers can also become attached to prototypes they have spent a great deal of effort producing; this can lead to problems like attempting to convert a limited prototype into a final system when it does not have an appropriate underlying architecture.

- ✓ Excessive development time of the prototype: A key property to prototyping is the fact that it is supposed to be done quickly. If the developers lose sight of this fact, they very well may try to develop a prototype that is too complex.

- ✓ A common problem with adopting prototyping technology is high expectations for productivity with insufficient effort behind the learning curve. In addition to training for the use of a prototyping technique, there is an often overlooked need for developing corporate and project specific underlying structure to support the technology. When this underlying structure is omitted, lower productivity can often result.

1.4.3 Incremental Development

According to Agile Development Tools (2010) the incremental build model is a method of software development where the model is designed, implemented and tested incrementally (a little more is added each time) until the product is finished. It involves both development and maintenance. The product is defined as finished when it satisfies all of its requirements. This model combines the elements of the waterfall model with the iterative philosophy of prototyping.

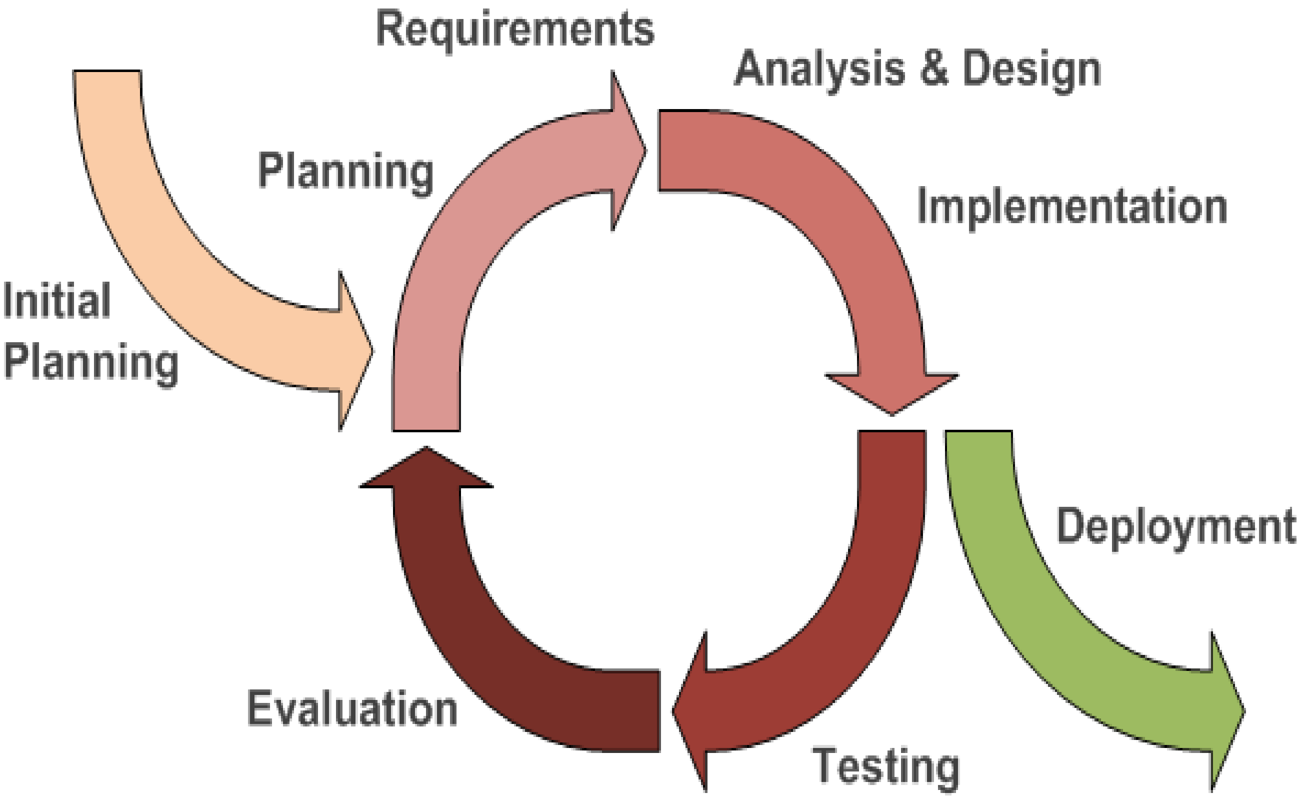


Figure 3: Iterative development (Agile development Tools, 2010)

Advantages:

- ✓ After each iteration, regression testing should be conducted. During this testing, faulty elements of the software can be quickly identified because few changes are made within any single iteration.
- ✓ It is generally easier to test and debug than other methods of software development because relatively smaller changes are made during each iteration. This allows for more targeted and rigorous testing of each element within the overall product.

Disadvantages:

- ✓ Resulting cost may exceed the cost of the organization.
- ✓ As additional functionality is added to the product, problems may arise related to system architecture which was not evident in earlier prototypes.

1.4.4 Spiral Development

According to Biko technology (2009) the spiral model is a software development process combining elements of both design and prototyping-in-stages, in an effort to combine advantages of top-down and bottom-up concepts. This model of development combines the features of the prototyping and the waterfall model. The spiral model is intended for large, expensive and complicated projects.

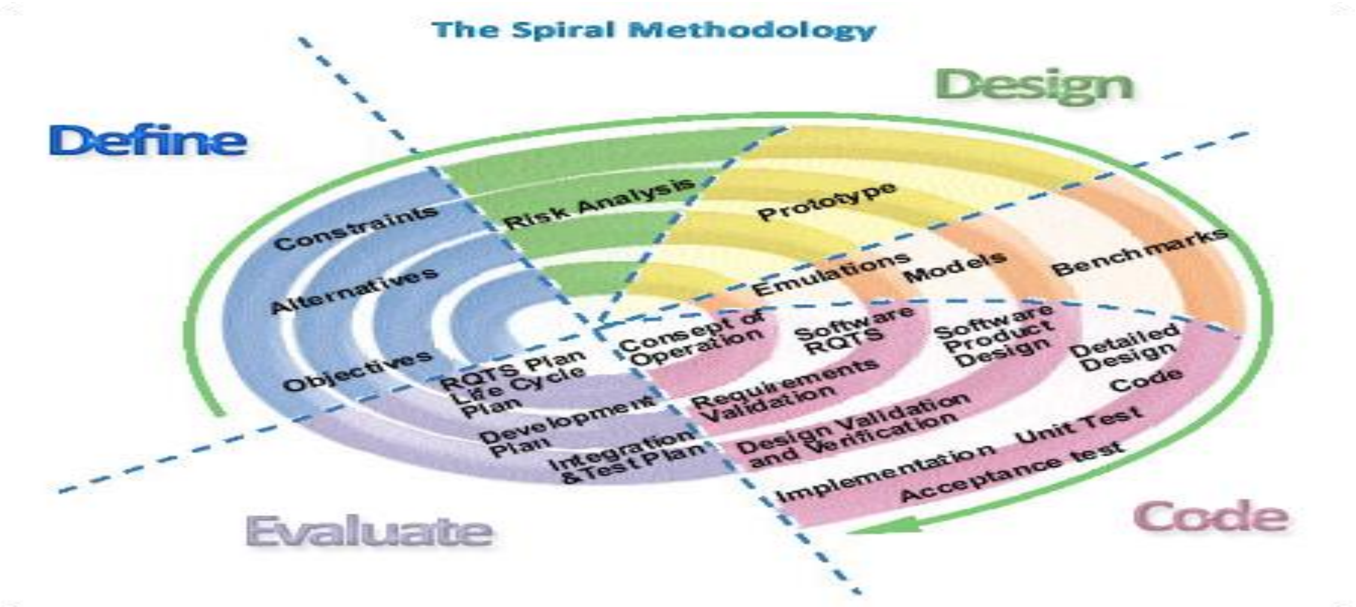


Figure 4: Spiral development(Biko technologies, 2009)

Advantages:

- ✓ Spiral Life Cycle Model is one of the most flexible SDLC models in place. Development phases can be determined by the project manager, according to the complexity of the project.
- ✓ Project monitoring is very easy and effective. Each phase, as well as each loop, requires a review from concerned people. This makes the model more transparent.
- ✓ Risk management is one of the in-built features of the model, which makes it extra attractive compared to other models.
- ✓ Changes can be introduced later in the life cycle as well. And coping with these changes is not a very big problem to deal with for the project manager.

Disadvantages:

- ✓ 1 Cost involved in this model is usually high.
- ✓ It is a complicated approach for big projects.
- ✓ Skills required, evaluating and reviewing project from time to time, need expertise.
- ✓ Rules and protocols should be followed properly to effectively implement this model. Doing so, through-out the span of project is tough.

1.4.5 Rapid application development

According to SpiraLabs rapid application development (RAD) is a software development methodology that uses minimal planning in favor of rapid prototyping. The "planning" of software developed using RAD is interleaved with writing the software itself. The lack of extensive pre-planning generally allows software to be written much faster, and makes it easier to change requirements.

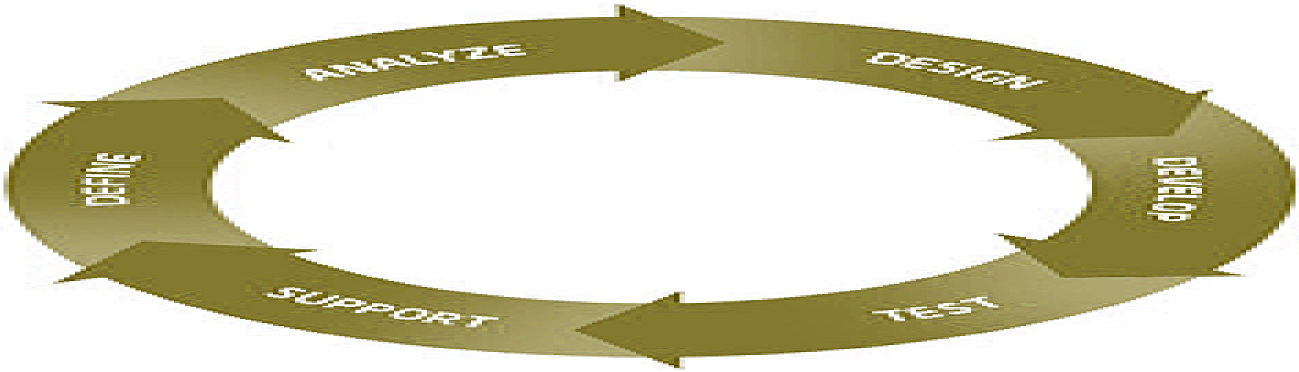


Figure 5: Rapid application development (SpiraLabs, 7 August 2010)

Advantages:

- ✓ Reduced development time.
- ✓ Increases reusability of components
- ✓ Quick initial reviews occur
- ✓ Encourages customer feedback
- ✓ Integration from very beginning solves a lot of integration issues.

Disadvantages:

- ✓ Depends on strong team and individual performances for identifying business requirements.
- ✓ Only system that can be modularized can be built using RAD
- ✓ Requires highly skilled developers/designers.
- ✓ High dependency on modeling skills
- ✓ Inapplicable to cheaper projects as cost of modeling and automated code generation is very high.

1.4.6 Agile Software Development(Scrum)

According to Hundhausen (2012, p.3), scrum is a framework for developing and sustaining complex products. Software is a complex product. Scrum is ideal for managing the development of software. Scrum is not a methodology or a process, although you can employ various processes within it. Software development doesn’t generate the same output every time, given a certain input. Scrum embraces this fact and is empirical, which means that it promotes the use of observation and experimentation in order to inspect and adapt. This enables a team to regularly see the effectiveness of its development practices and make changes accordingly.

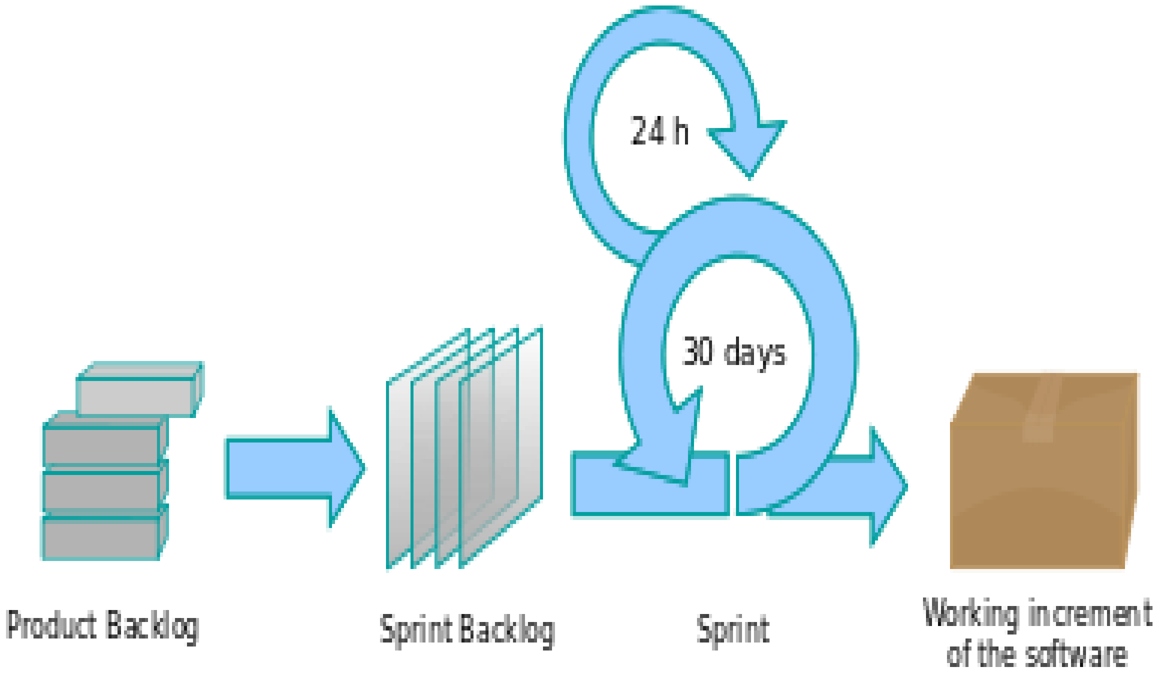


Figure 6: Scrum development (Agile development tools 2010)

Advantages:

- ✓ Managing the development of software.
- ✓ Scrum uses empirical approach, which has the privilege of observation and adaptation.
- ✓ The team is able to make effective self-analyses of the development and make changes accordingly.

Disadvantages:

- ✓ In case of some software deliverables, especially the large ones, it is difficult to assess the effort required at the beginning of the software development life cycle.
- ✓ The documentation and the design could be underestimated.
- ✓ The senior employees are able to make decisions how much time a specific task is going to take. A team of only new employees is not a good practice of using scrum.

1.4.7 Chosen Methodology

After carefully considering all 6 types of methodologies, decision has been made to choose the agile software development (scrum). This has been chosen because it is easy to respond to any changes, customers gets the most important work first, the work that is done will be better accepted by the customer and improved productivity. Working with scrum appealed most to me as a semiprofessional software developer, this was the most practical choice for me. It fits in to the work as it mostly concentrates on the Analysis and Design of software development system and in this way it will take less time to do the whole system.

1.5 UML

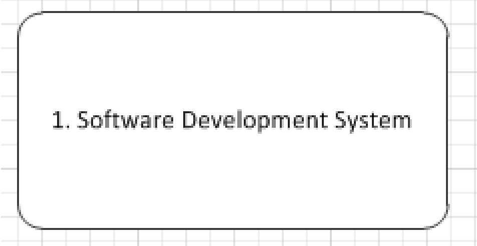
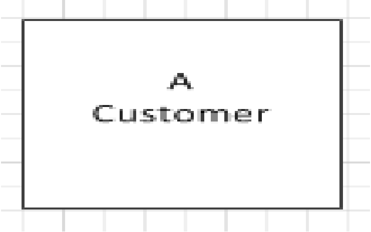
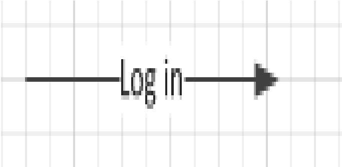
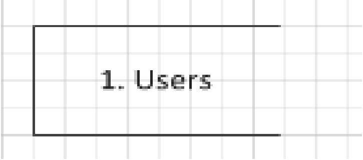
According to UML definition (2011) UML is short for Unified modeling language and is an ISO (international standard) specification language for modeling objects. It's a refinement of earlier object orientated design and object orientated analysis methodologies. The focus will be on creating 5 different types of diagrams to show how the system works. The 5 types are going to be: activity diagram, use case diagram, class diagram, sequence diagram and a state diagram. A key will be included within the UML section to help the reader understand fully what the diagrams represent.

1.6 Web design and implementation

As there is going to be created a website for the system, it has to be done properly. The focus will not be to plan and analyze sections as they will have been covered within this documentation. However, the design of the website needs to be done. It will not be necessary to build and maintain the website because the time does not allow and this is not the purpose of the document. The main sections within the design stage are the structure, visual style, navigation and pages. All these will be covered in detail in the Web design and implementation section (Heading 4).

1.7 Data Flow Diagram Key

To Structure the Context diagram and the Data flow diagrams Microsoft Visio Gene-Sarson (Metric) Notation will be used. The first phase is the context diagram which labels the users interaction with the system. The data flow diagram offers a more in depth understanding of the individual processes from the context diagram. Here are the diagram objects definition and visual to help understand both the context diagram and the data flow diagrams.

Process – the Process symbol represents the processes which happen within the system.	
Entity – The entity symbol has an input or an output in to the system. The entity does not exist within the system and in our case represents the customers and what type of interaction they have with the system.	
Data flow – The data flow represents data movement within the system. An arrow represents the direction of flow.	
Data Store – The data store symbol is used to show the location of the data base.	

2. Analysis Specification

2.1 Terms of reference

The terms of reference section gives us an introduction to the project situation. It will outline what the proposed system will do and achieve. The documents within the terms of reference are as follows: Statement of purpose, User requirements, target audience and restrictions and constraints. Each of these documents will give a brief understanding of what they explain. The systems analysis on a whole determines the interactions within the system.

2.1.1 Statement of Purpose

The system is intended to attract individuals, companies and organizations to use its services. So the purpose of the system is to provide the clients with a service, software development that is easy to use, fast to implement and on a competitive price. The system will allow its customers to contact them and discuss the details of the projects. The full aspect of the system will be explained in the requirements. As systems of this specification are already available, I will have to research in to them to see what makes their systems so successful and what our systems can offer that theirs cannot.

2.1.2 User Requirements

The user requirements show what the systems can accomplish. The requirements show all the possible scenarios the user can encounter within the systems. The list will **NOT** tell the user how to do them, only what the systems will do.

1.1 Non- registered clients will be able to see the sample project the company has done so far.

1.2 Non- registered clients will be able to see the price list of sample projects.

1.3 Non- registered clients will be able to contact the company and require additional information.

1.4 Non- registered clients will be able to register.

2.1 Registered clients will be able to book an appointment for consultation.

2.2 Registered clients will be able to explain their project and get an approximate price about it.

2.3 Registered clients will be able to recommend a friend to the company and get a bonus (£50).

2.4 Registered clients will be able to get a discount for being loyal members.

2.5 Registered clients will be able to give feedback for using the services of the company.

2.1.3 Target Audience

The aim of the system is to attract customers from different backgrounds like private companies, organizations and individuals to get their desired software developed. The system will encourage the current customers to advertise the services of the system. So when one customer is happy with the services of the company he will definitely recommend it to his friends which are potential customers. The system will also advertise its services in the social network in places like Facebook, Twitter, LinkedIn and others.

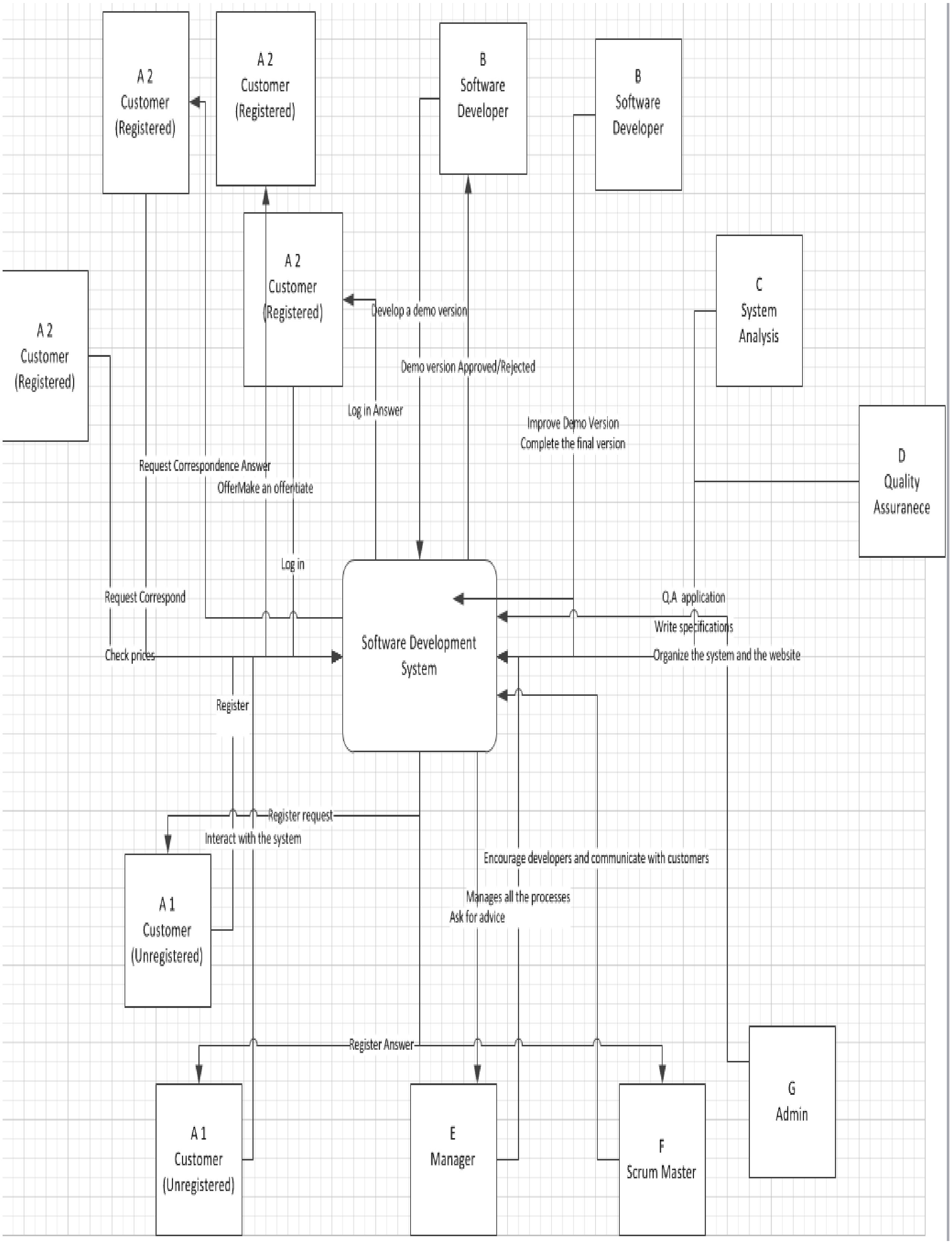
2.1.4 Restrictions and constraints

During the project it will be reached a point where it will be hardly possible to advance. This will be due to restrictions and constraints. Here are a few that will be encountered while analyzing and designing the system. The software: As the work is going to be done from home, it will not be able to access the high quality pieces of kit that professionals use in order to design and create their systems, because the software is very expensive and it is morally wrong for me to use unlicensed version of it.

The time allocated to complete: As this is a university project, there is a certain timeframe to complete the project. This is due to the marking phase the project needs to go under in order for it to receive a remarkable grade. This means there may not be enough time to fully maximize the ideas towards the project and the output may not be as good as it could be. As well as marking, depending on the amount of functions the system will have, the longer it will take to add to the project, so a system will have to be designed that can be built to fully maximise the main functions. A home PC will be used to install Microsoft Visio (for the data flow diagrams) and Star UML for my UML diagrams. ASCENT is not accessible to me due to my 64 bit Windows 7 machine, so it will be compromised and used Microsoft Visio to draw the diagrams. This isn’t the best way to draw diagrams, but it can provide a decent diagrams.

2.2 Context Diagram

The context diagram was constructed to identify the actors outside the system which interact with the system. These actors will interact with the system regularly and the black arrows symbol data flow, alongside with a description. The actor will normally enter something in to the system, and the system will generate data out of the system, or the other way round. This diagram is the highest level view of the system. Further diagrams such as a top level data flow diagram and a lower level data flow diagram will go in to more detail about the processes of the system.


A 2
Customer
(Registered)
A 2
Customer
(Registered)
B
Software
Developer
B
Software
Developer
A 2
Customer
(Registered)
A 2
Customer
(Registered)
Develop a demo version
Demo version Approved/Rejected
C
System
Analysis
Log in Answer
Improve Demo Version
Complete the final version
D
Quality
Assuranece
Request Correspondence Answer
OfferMake an offertiate
Log in
Request Correspond
Q.A application
Write specifications
Check prices
Software Development
System
Organize the system and the website
Register
Register request
Interact with the system
Encourage developers and communicate with customers
A 1
Customer
(Unregistered)
Manages all the processes
Ask for advice
Register Answer
A 1
Customer
(Unregistered)
E
Manager
F
Scrum Master
G
Admin

2.3 Events List

The events list provides a text based list of the processes and events the system needs to respond too concerning the actors (Customer, Software Developer, System Analysis, Quality Assurance, Manager, Scrum Master and Admin).

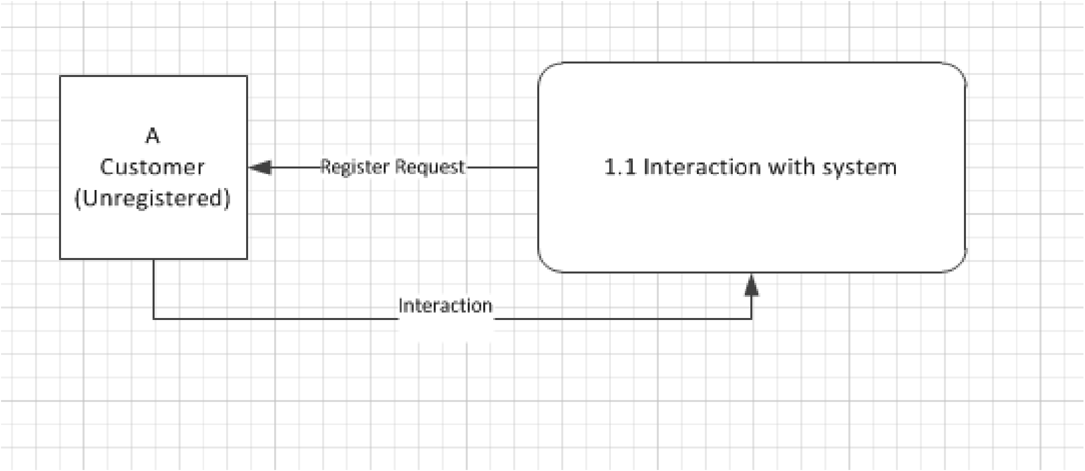
1.1 1.2	Customer(Unregistered) Customer (Registered)
2 3 4 5 6 7	Software Developer System Analysis Quality Assurance Manager Scrum Master Administrator
Event list	
1.1.1 1.1.2	Interact with the system Register
1.2.1	Customer looks at the example price ranges for specific software development
1.2.2	Customer request a meeting Online/Personal
1.2.3 1.2.4	Customer considers an offer / negotiate Log in
2.1	Software Developer makes a demo version
2.2	Software Developer Improve the demo
2.3 3.1 4.1 5.1 6.1 7.1	Software Developer works on completion the final version System Analysis Write the technical Specifications QA Test the application Manager Give work and general help. Scrum Master encourages employees and communicates with customers. Administrator organizes the system and the website.

2.4 Data Flow Fragments

For each point in the event list these data flow fragments will be constructed. The fragments convert the events (inputs in to the system) and create processes. Those processes deal with the terminators and their inputs and outputs (creating data flow). The notation for data flow is a directional black arrow (where the data is heading) and it will normally come with a short description telling the reader what the flow means. If the processes require a storage space for data, a database like function is created. These functions are called data stores. The data stores need to be written and read at least once in the data flow diagrams to have effect. Each fragment is not going to be described, as it has already been described in the top level and lower level diagrams in greater details.

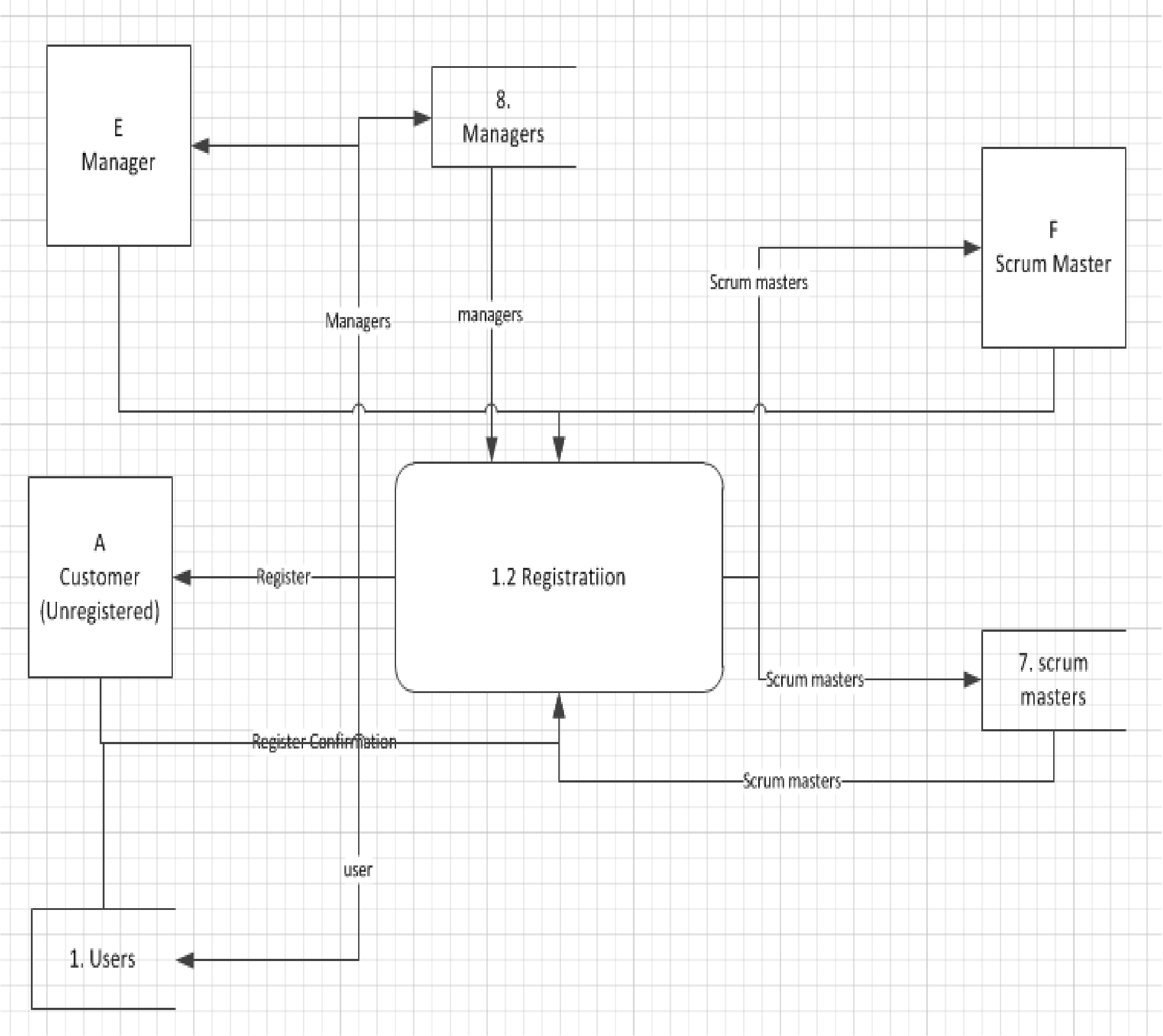
2.4.1 Interaction with the system

The first data flow fragment comes from the first event on my event list Unregistered user request to interact with registered part of the system. The fragment shows that the terminator (an unregistered user) interacts with the system. This is shown as an arrow indicating a data flow. The direction of the data flow indicates whether it is an input or an output. The terminator inputs their interaction into the system and the system automatically outputs a registration request to the terminator, requesting that they register to view that section of the system.



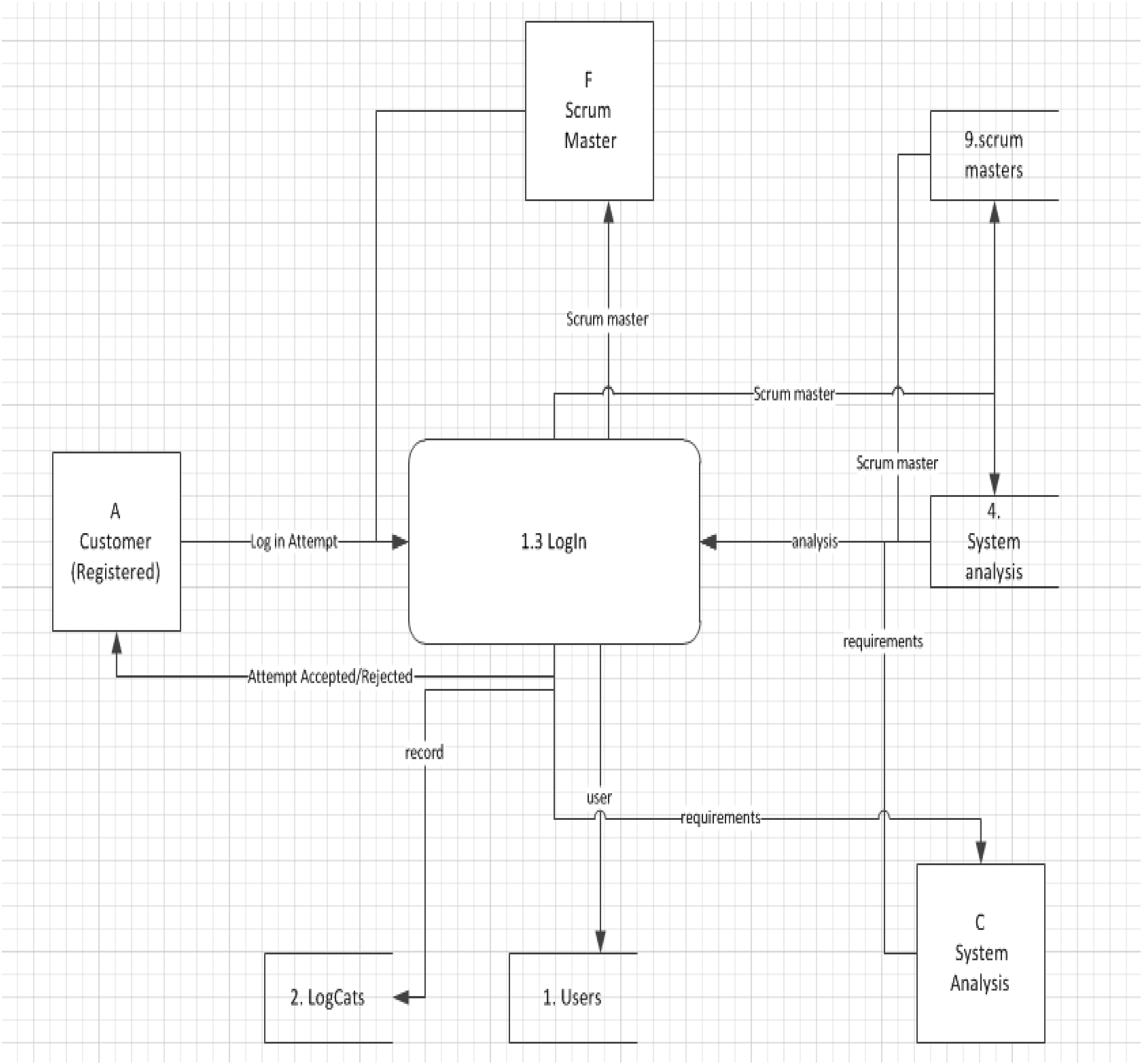
2.4.2 Register

Once an unregistered customer has received a registration request, they will have to register in order to view the user only content on the system (website). The customer will enter their details into the system. This is input from the unregistered customer. The system will send a registration confirmation to the unregistered customer to confirm that they have just became a customer. The information the customer has just inputted into the system is stored into the database for the customers.



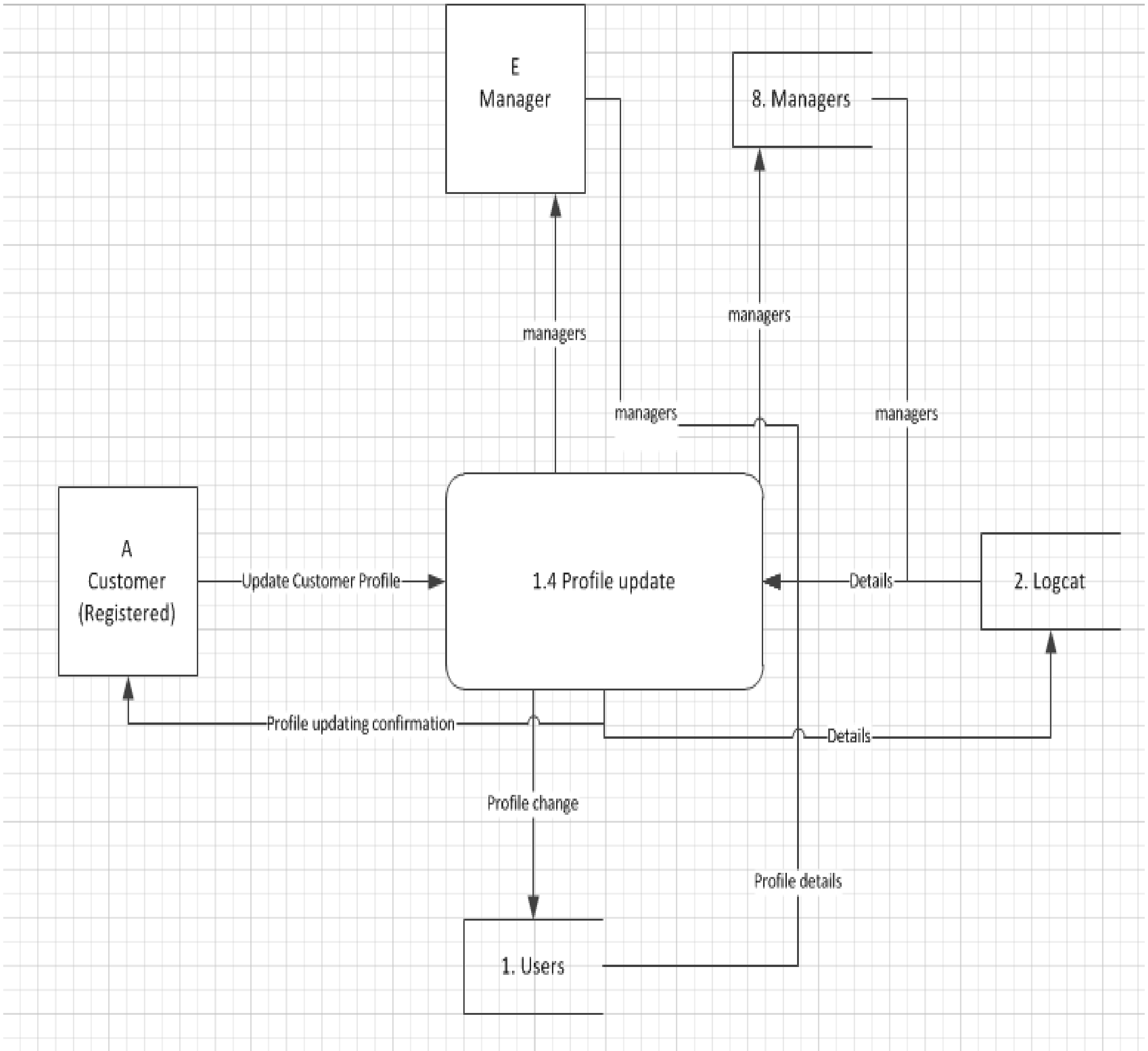
2.4.3 Customer Log in

The log in fragment shows how the customer will log in. The terminator status has changed as the unregistered customer has now registered. The new customer attempts to log in with their new username and password they created in the registration stage. If the customer has correctly inputted their username and password, they will gain access in to the system. The system then outputs the accept/reject message to the user. The user store is needed as the system needs to cross reference the user's username and password. The success/failure of the log in is then written in to the log in audit data store.



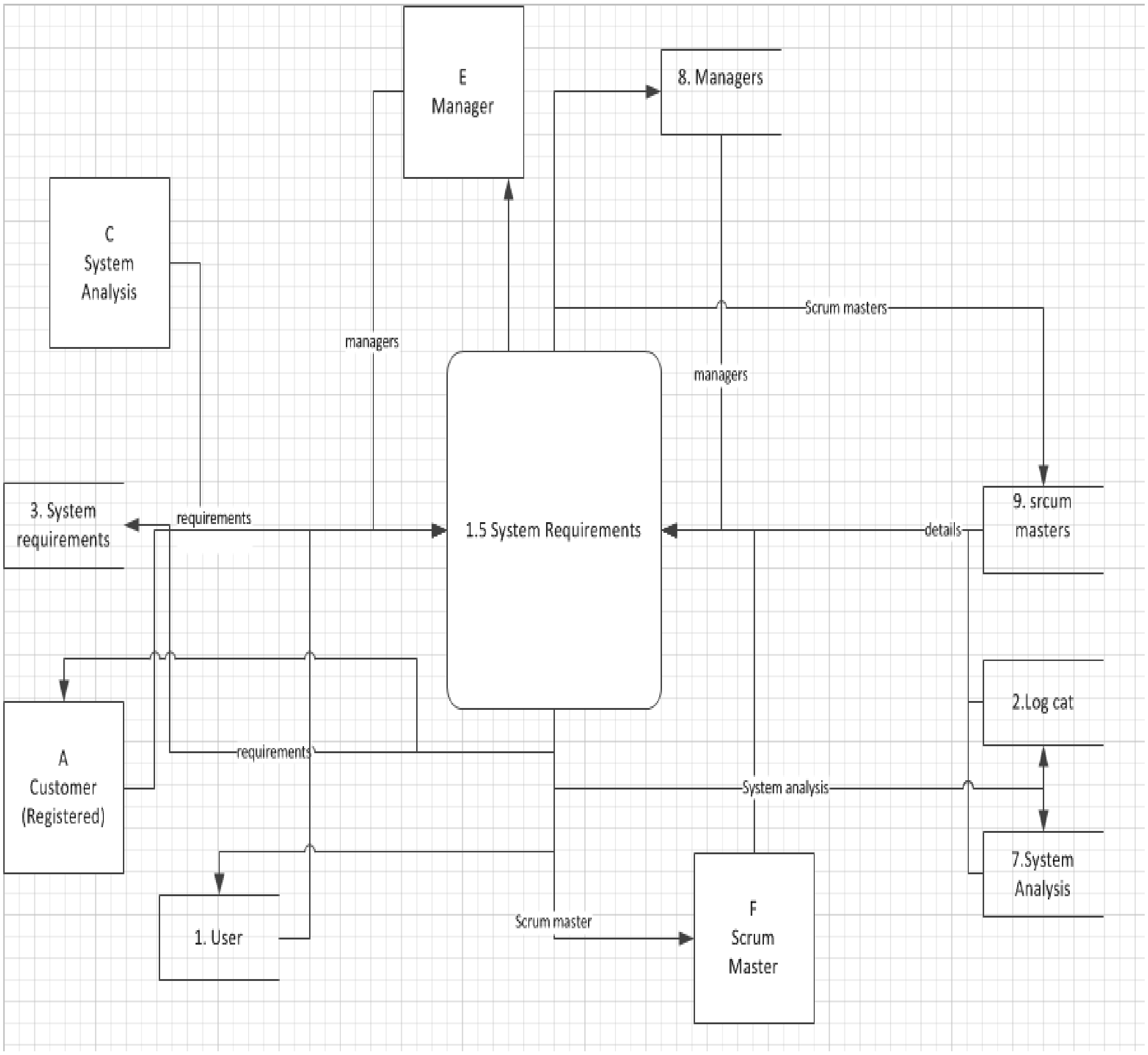
2.4.4 Profile update

The profile update fragment deals with the user updating their profile while logged in to the system. The user will input the changes to their profile and the system will automatically change the profile and send the user a confirmation as an output. The profile changes are written in to the user data store and the profile details are read out of the data store so the user can see their new changes in affect.



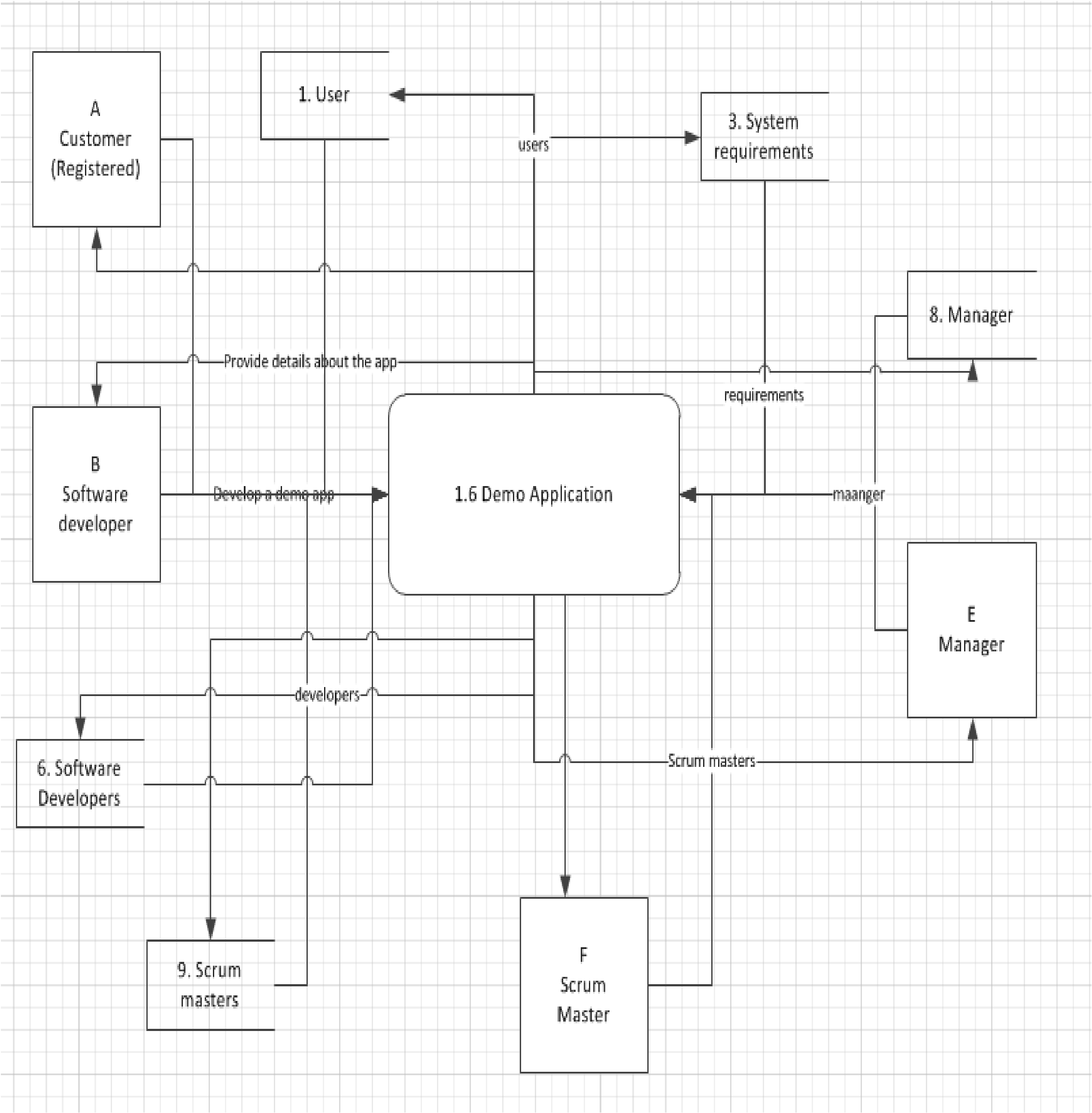
2.4.5 System Analysis information process

The system analysis will be responsible for the interaction with the customers. He will collect the information that is necessary about the application. Then he will filter it and design it in appropriate way for the software developers to implement is. For example he will make class diagrams which are the backbone for the software development.



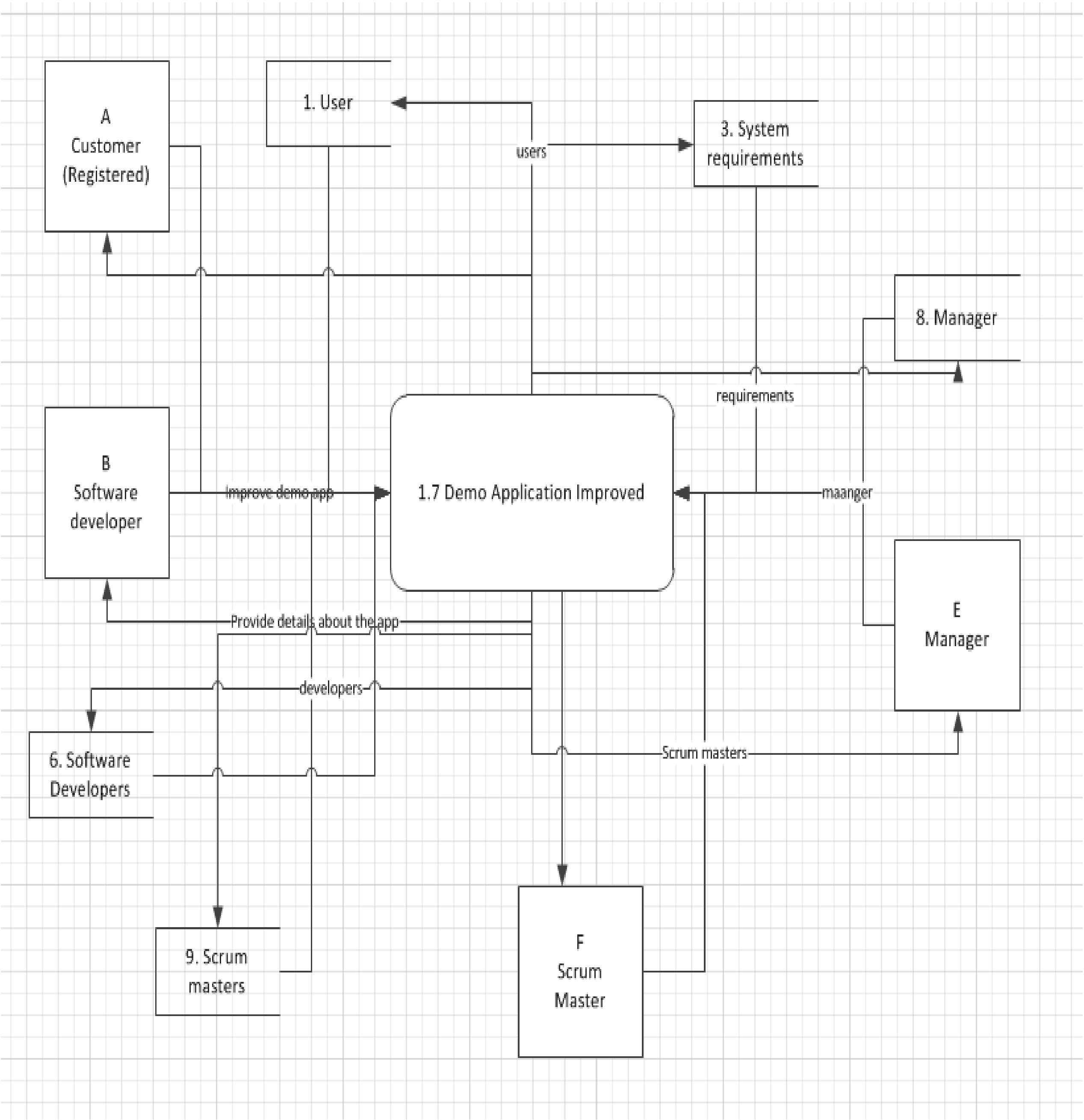
2.4.6 Software Developer demo application

This will be the main fragment which involved the software developer. The software developers will be responsible for development of a demo version of the application. The software developers will have a ready class diagrams, use case diagrams and other uml diagrams. The demo will need to be done into the shortest period of time as possible. Then the customer is going to review it and express his/her opinion about it.



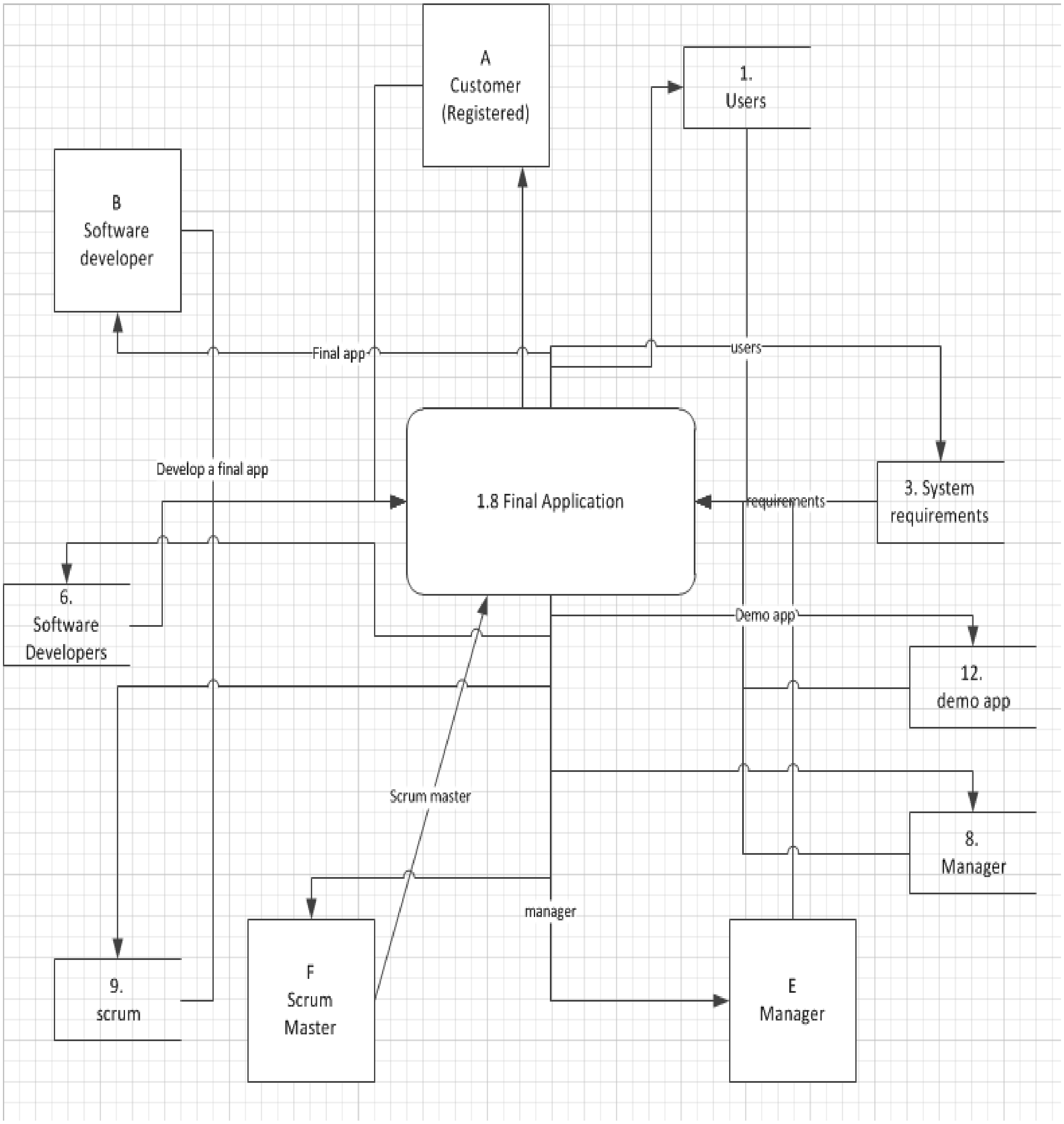
2.4.7 Software Developer Improve the demo

After the development of the demo application is finished it will be showed to the customer. The customer will judge its functionalities, abilities and will make a statement. If the customer does not like the demo application, than the software developers together with the whole team will have short period of time to make it appropriate and desirable.



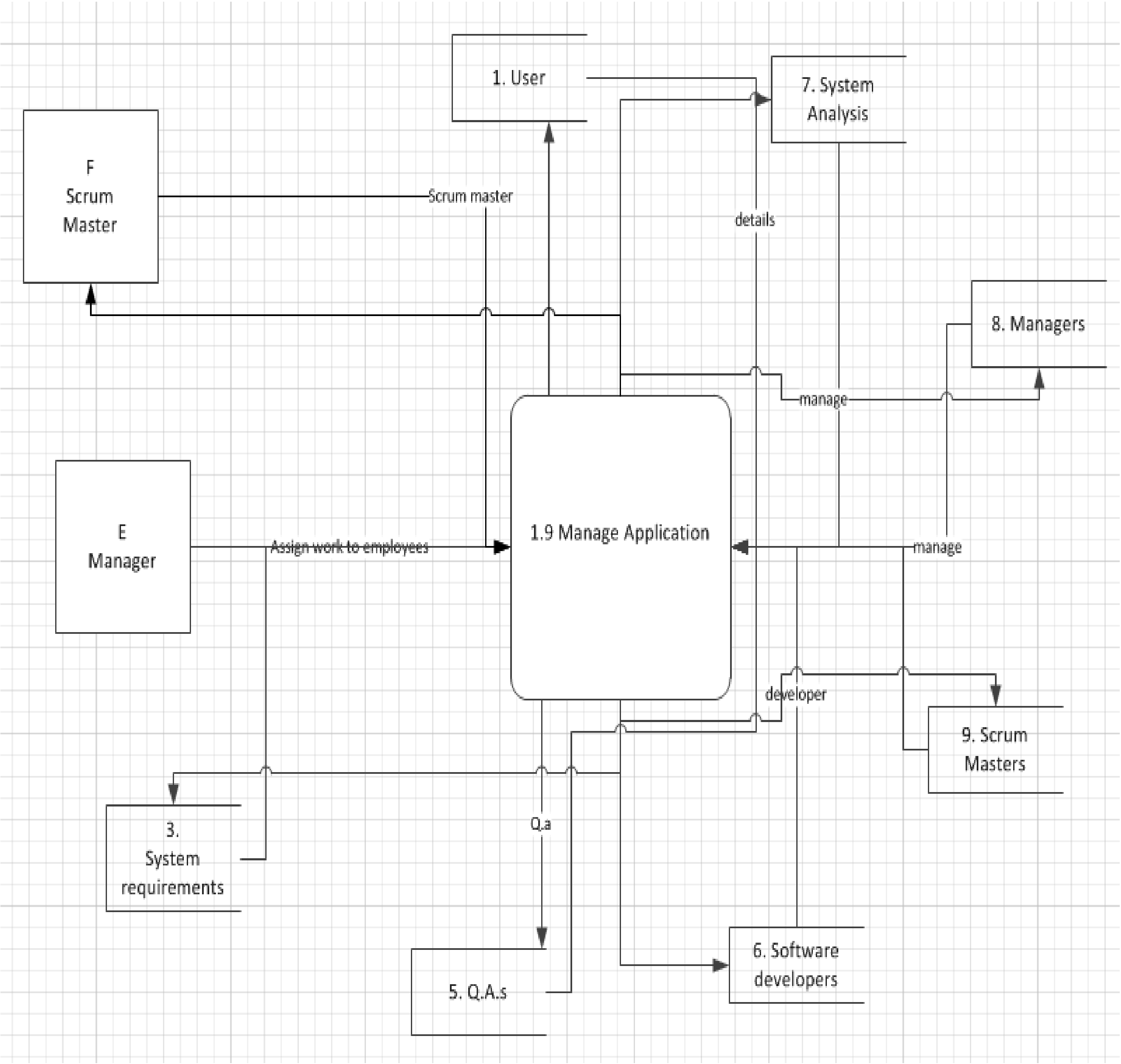
2.4.8 Software Developer final version

After the development of the demo application is finished it will be showed to the customer. The customer will judge its functionalities, abilities and will make a statement. If the customer approves the demo, than the software developers will work on to make a final version of the application.



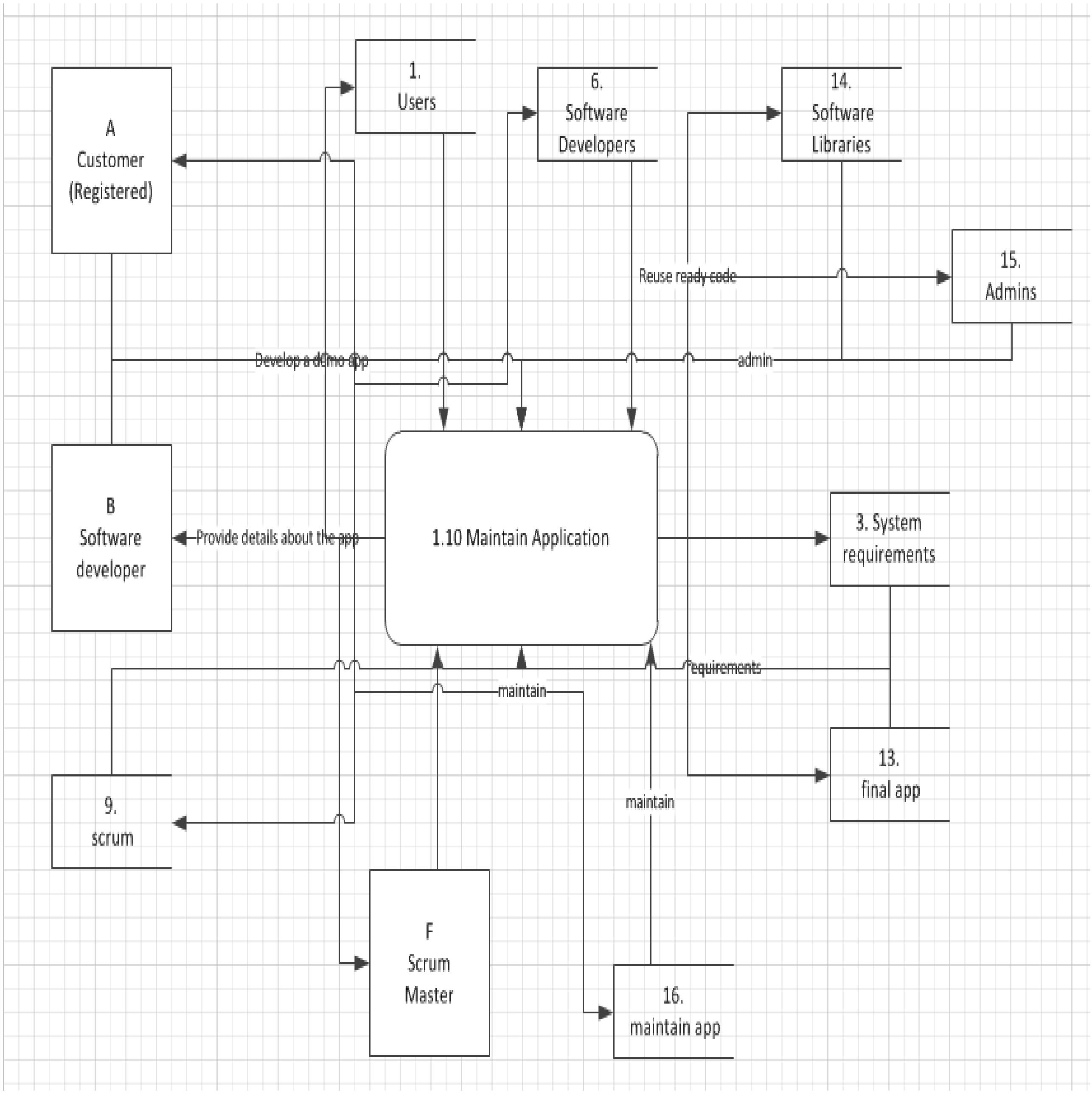
2.4.9 Manage the app

The main responsibilities of the managers are to assign work to all the employees. This way everyone will be busy and will work for the good of the system. Managers are also to keep that the employees are doing their jobs on professional level as expected.



2.4.10 Software Developer maintain the app

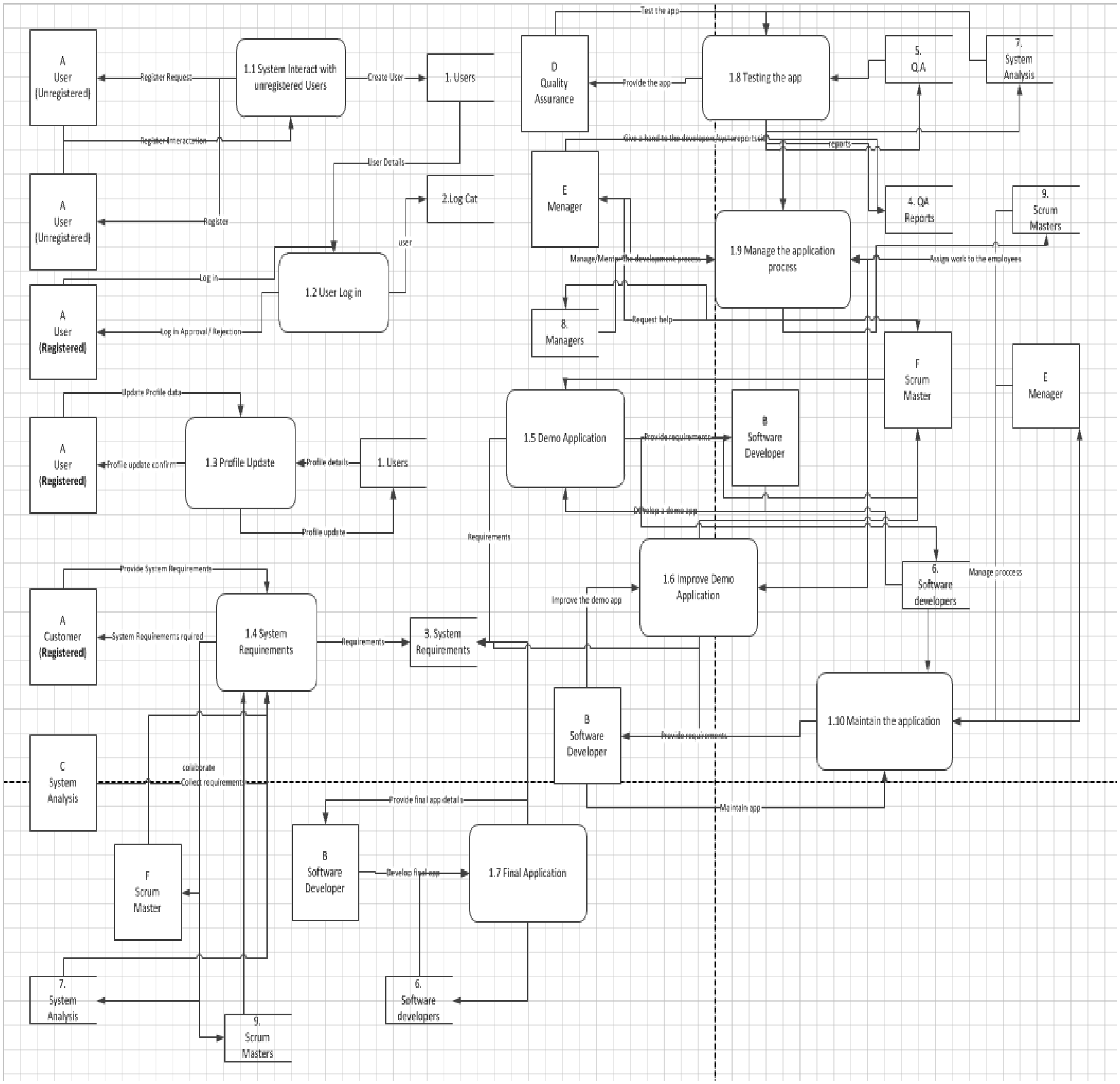
After the software development phase is finished the software developers are responsible for maintaining the software. This is a major task in their duties and is underestimated in a lot of companies. Because the software has been written by professional standards it is relatively easy to maintain it. However if the customers would like their software to be maintained by this system but the software has not been developed in professional standards another fees will apply.



2.5Data Flow Diagrams (DFD)

2.5.1 Top level DFD

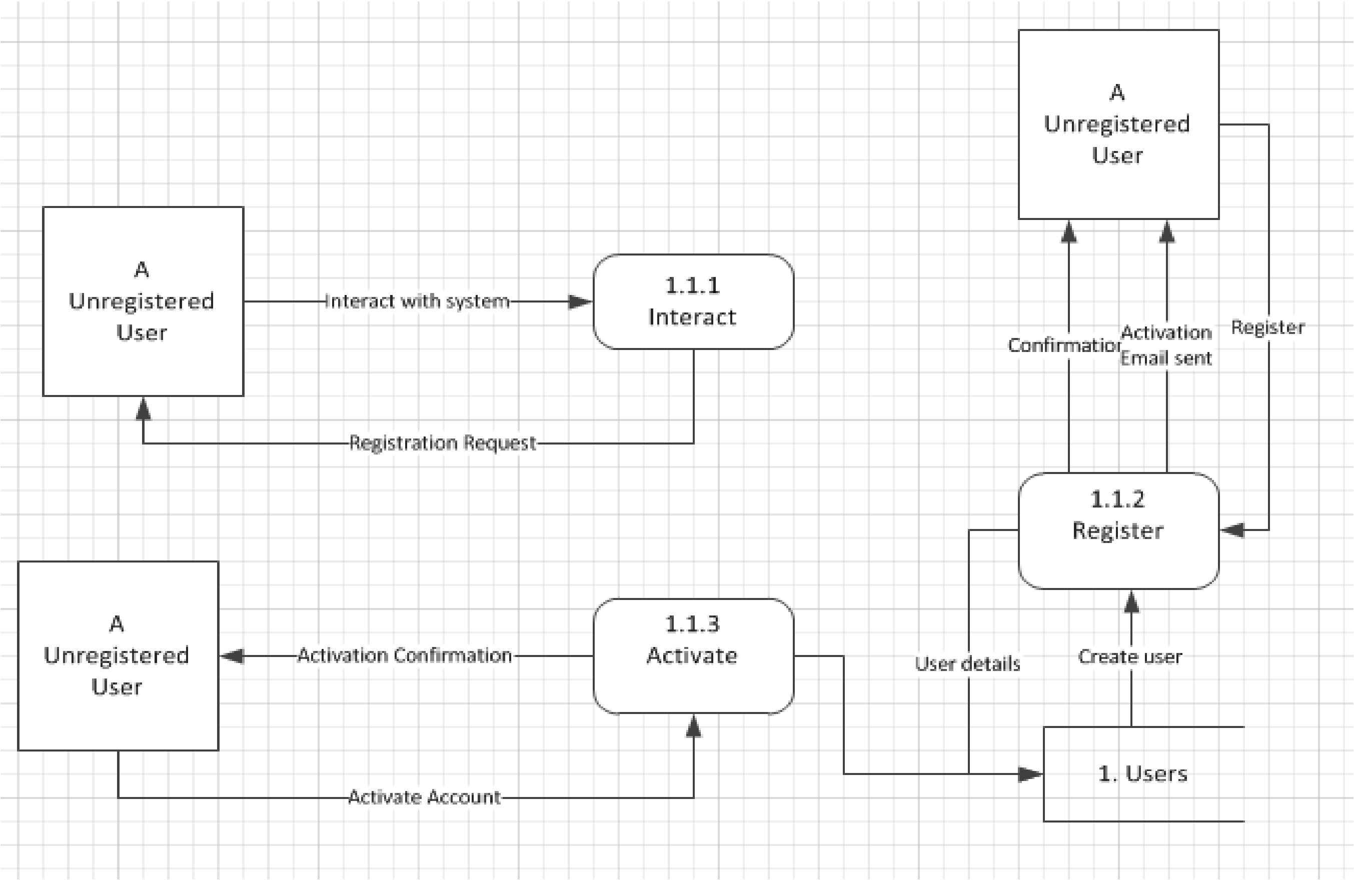
To construct the Top level data flow diagram I had to utilize the information from my context diagram and event list, while using Microsoft Visio. The top level data flow diagram is an exploded view of the context diagram and provides much more detailed view of the whole system. I will then go and explode each process further to go in to more detail. This will give much more understanding on how much system operates.



2.5.2Low level DFD

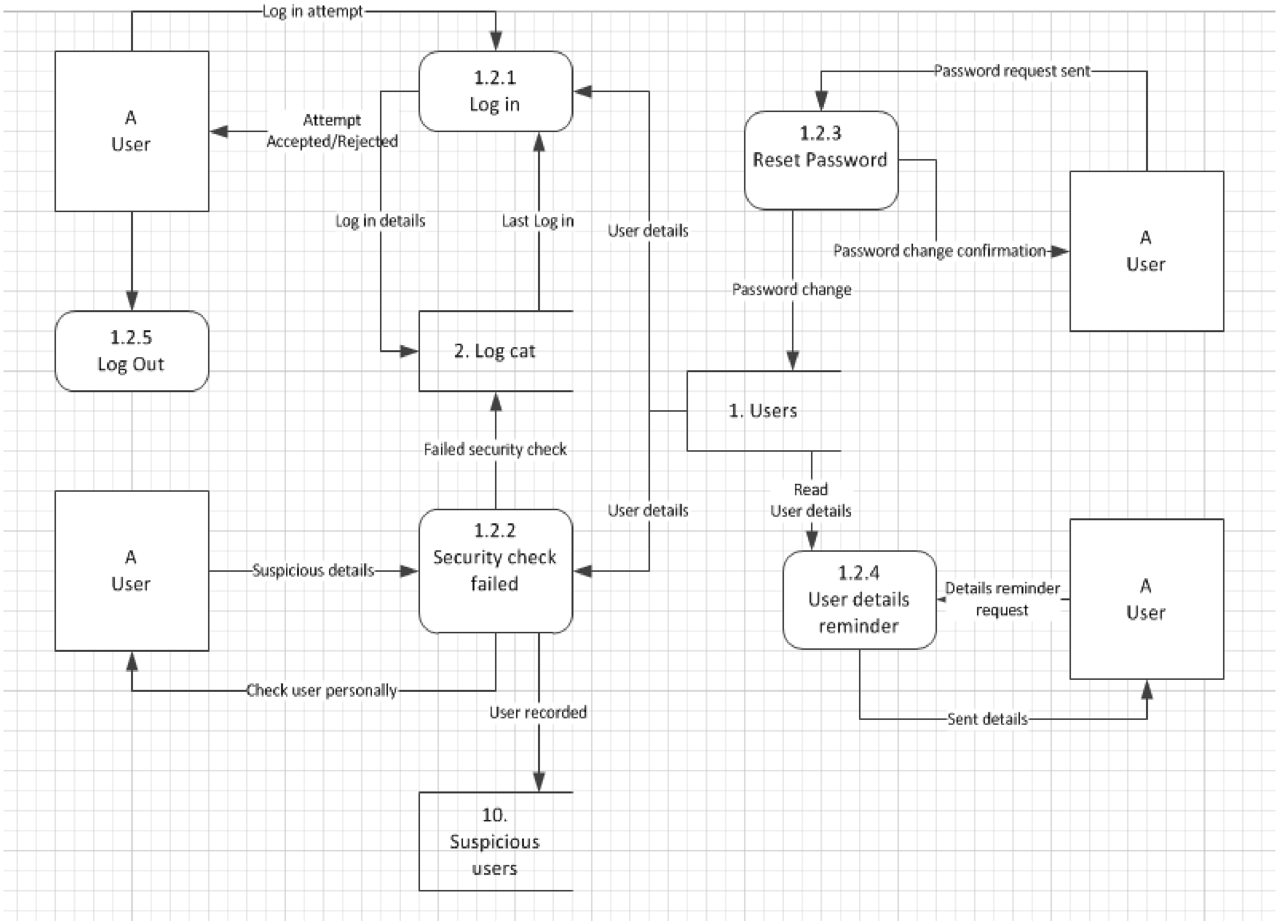
2.5.2.1 Manage Unregistered Users

The manage user process in the top level data flow diagram has been explained in more detail with this lower level data flow diagram. The lower level data flow diagrams go in to much more detail than the top level. However for this lower level diagram, I have only added the activation process. The activation process means that the new user has to activate their account in order to continue with the website. The register process writes the users details in to the data store. This is the first and most important data store as it will write/read in to many other processes



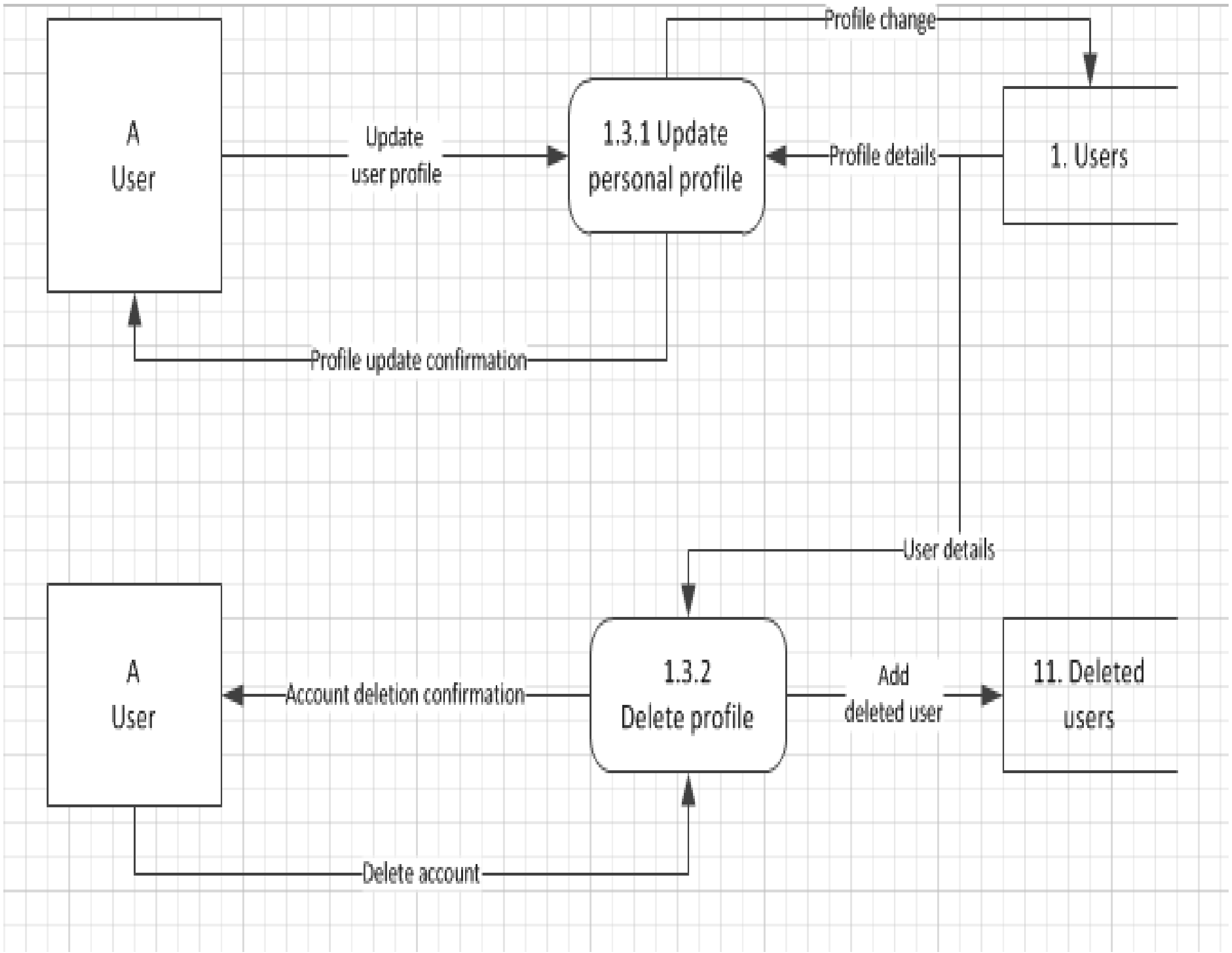
2.5.2.2 User Log in

The lower level data flow diagram for the log in process goes in to much more detail. The main processes are common for a log in system. It has been included a processes that help the user if they forget their log in details. The extra process that is included is the attempts exceeded. A lot more security I could have been included a when resetting the password, but that would mean going in to a lot more details and going in to further details for the Reset password process. Due to the length and time restrictions for this project, it was decided against it. There is a security check of the customers in order to prevent any frauds and if anything suspicious or extraordinary occurs during the check, the user goes to the data store for the suspicious users. This is done in order to reduce the risk of unstable users and users who try to take advantage of other users and the system.



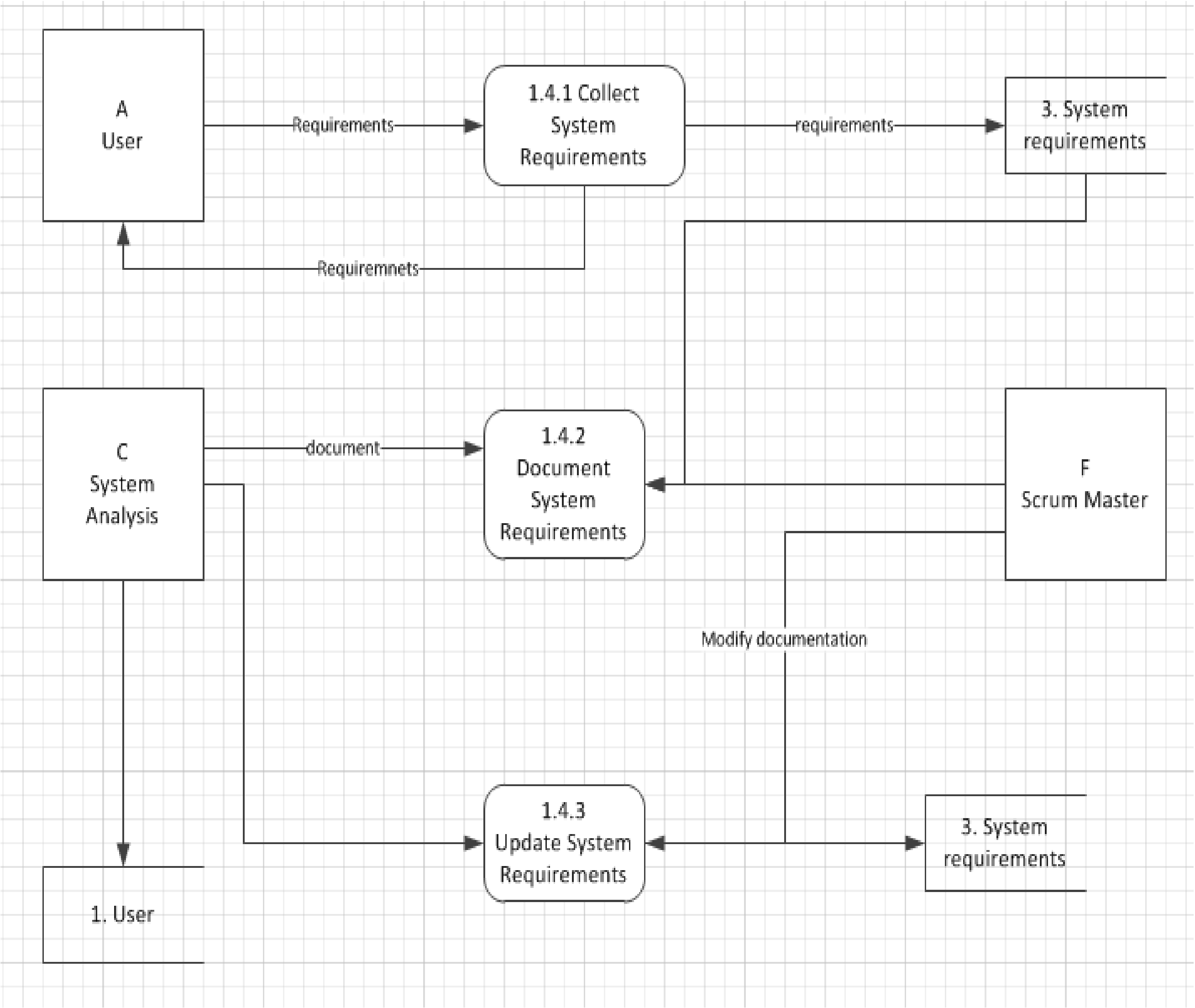
2.5.2.3 Profile update

This lower level data flow diagram is a detailed diagram of the manage profiles process. The diagram adds extra data stores which weren't involved in the top level diagram. The diagram is simple and only has 2 processes. The user can change their personal profile information like address, phone number, email and credit card number. All the personal information is going to be encrypted in order to prevent any data leak which will bring bad reputation to the company.



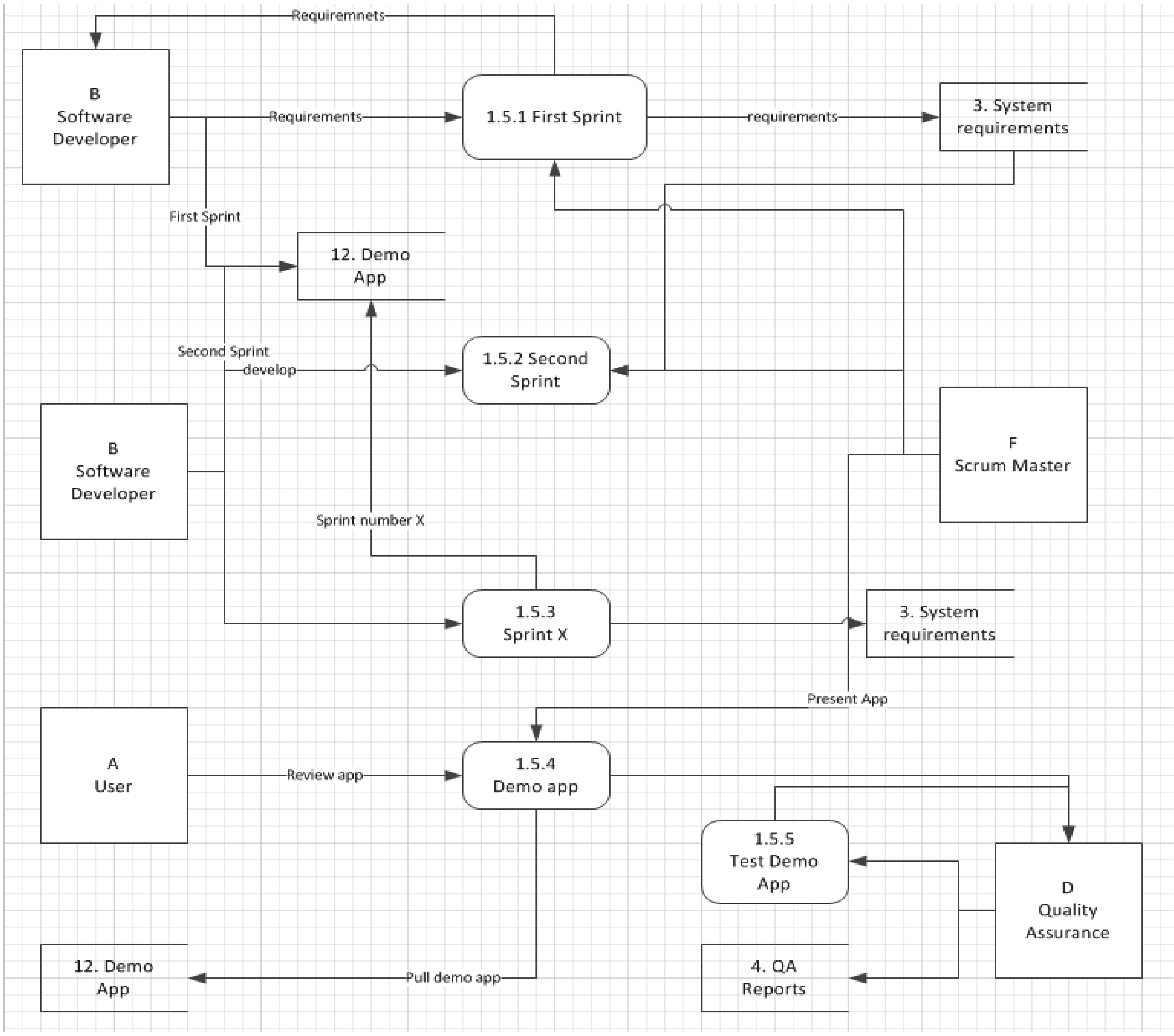
2.5.2.4 System Requirements

This lower level data flow diagram shows the processing of the system requirements. There are 3 main processes. First the user meets with the System Analysis to collect the requirements and document them. Once this is done the requirements are being documented and send to the user for review. If there are major inconsistences with the user expectation the Scrum Master helps the System Analysis to update the requirements.



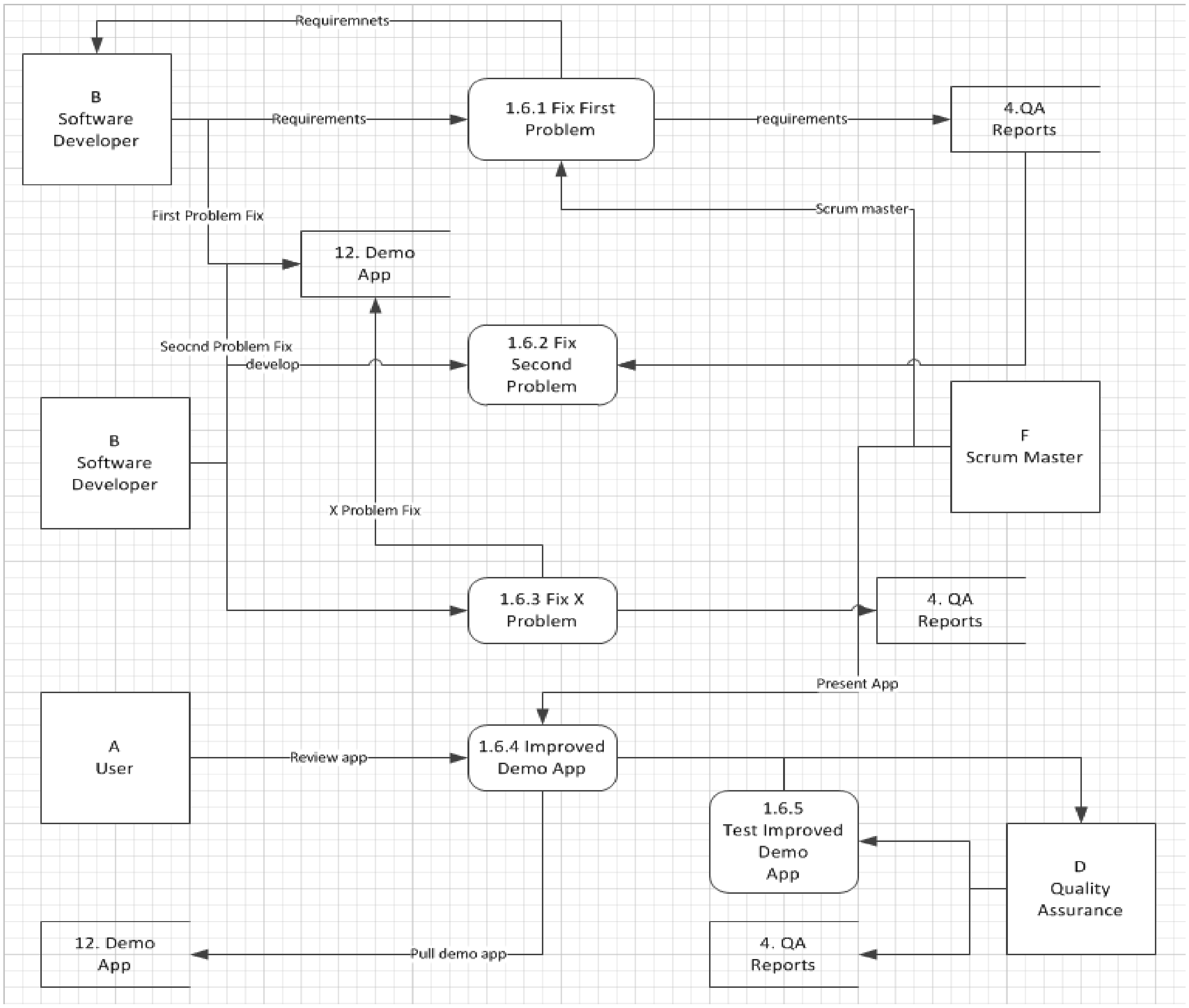
2.5.2.5 Demo Application

This lower level data flow diagram for the demo application describes the model for developing a demo application. First the Software developers pull the date from the system requirements data store and start doing the sprints. Usually one sprint is between 1 week and 1 month. When all the sprints are done then the QA test the application and if the tests fail then the software developers needs to improve the application otherwise the Scrum Master presents the Application to the Users.



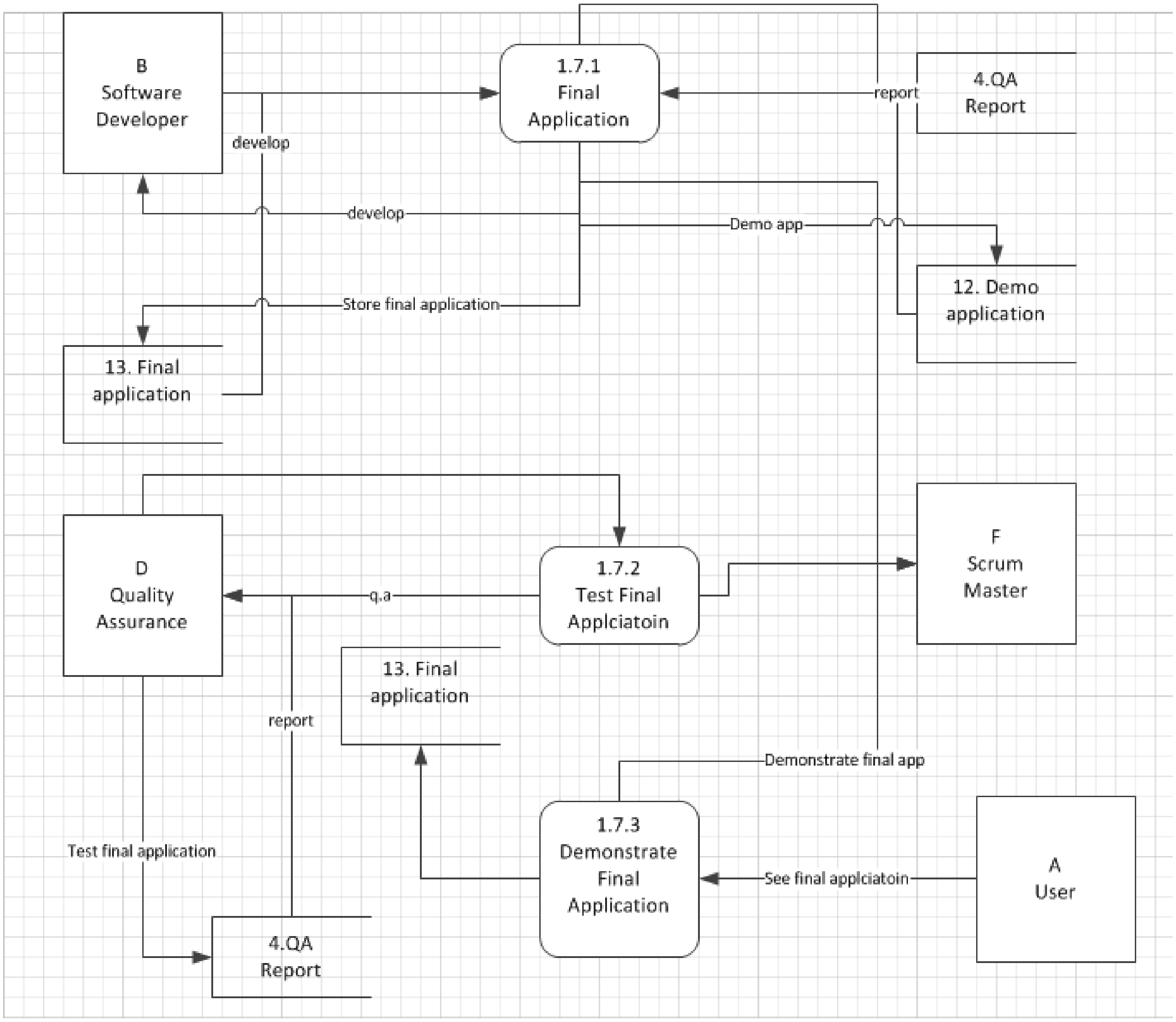
2.5.2.6 Improve Demo Application

The lower level data flow diagram for the improve demo app log in is very similar to the demo app. When the demo application is finished and major flow backs are found or the customer is not happy then the demo app is brought back to the software developers to improve it. This loop works until the demo application is good enough and usually there shouldn't be more than one improvement for the demo application because it is demo application and does not have all the features of the application so it will be spend too much time.



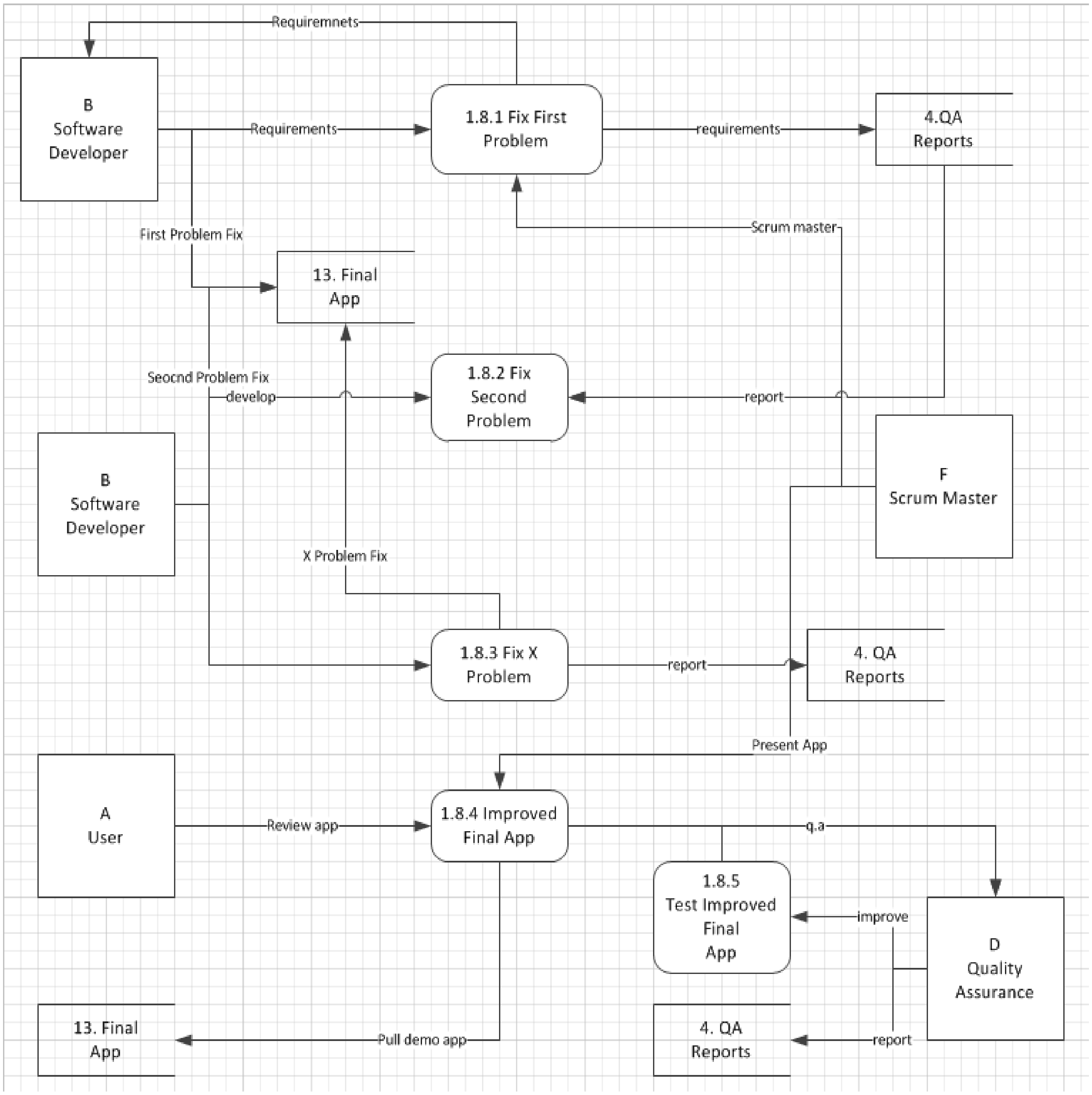
2.5.2.7 Final Application

This lower level data flow diagram is the exploded view of the top level data. The Software Developers takes the demo application and develop it until they have done all the functionality for the final application. After that the final application is tested by the quality assurances and if any significant problems are found then the final application is given back to the software developers to modify it. Once the final application is ready and tested the scrum master presents it to the clients/users and they go home happy.



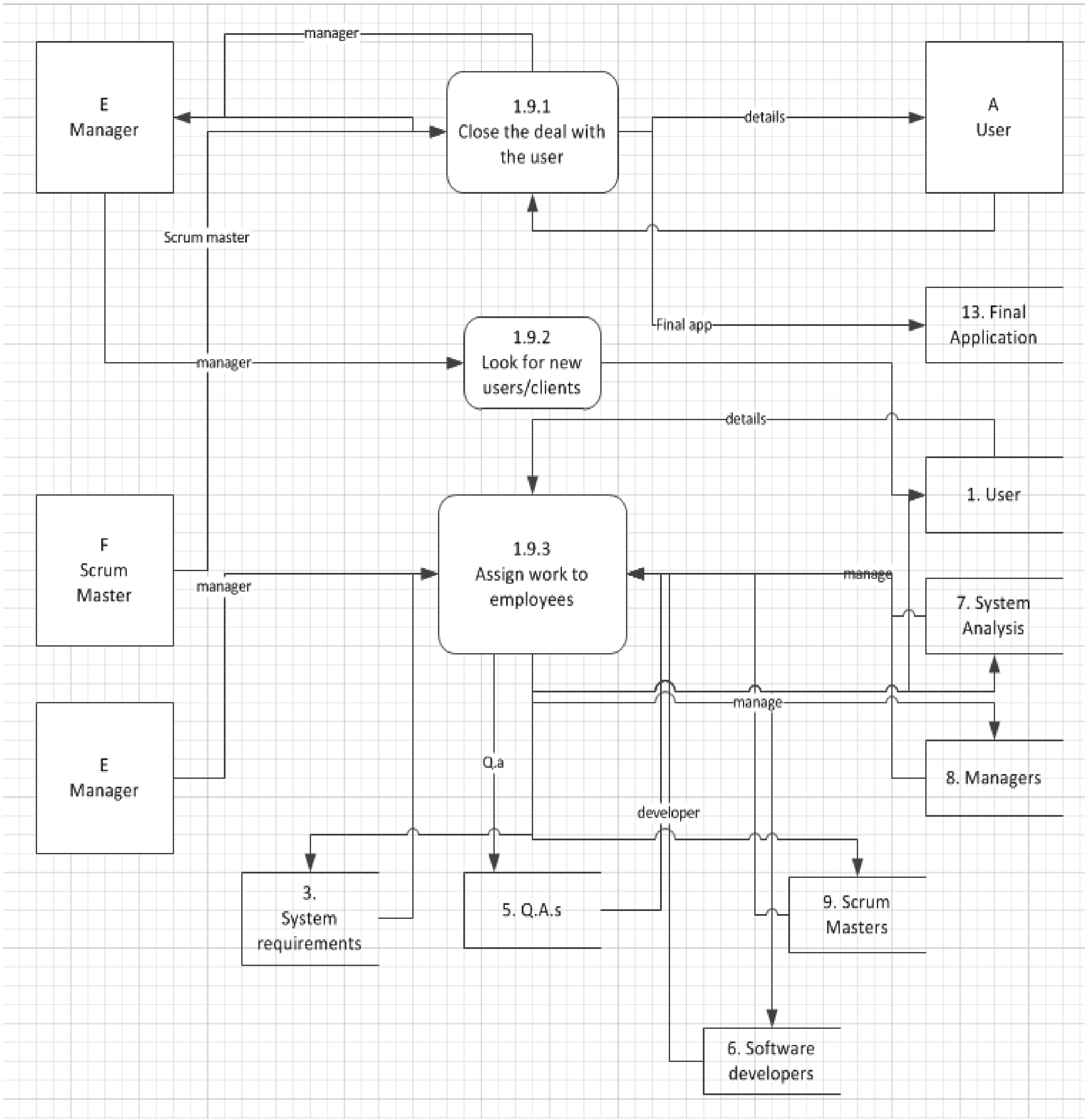
2.5.2.8 Test Final Application

This lower level data flow diagram for the test final application describes the model for testing the final application. In case the users/clients are not happy with the final application or the tests have failed the application is send back to the software developers to finish it. This process continues until both conditions are satisfied. Then the Scrum master presents the final application to the users/customers.



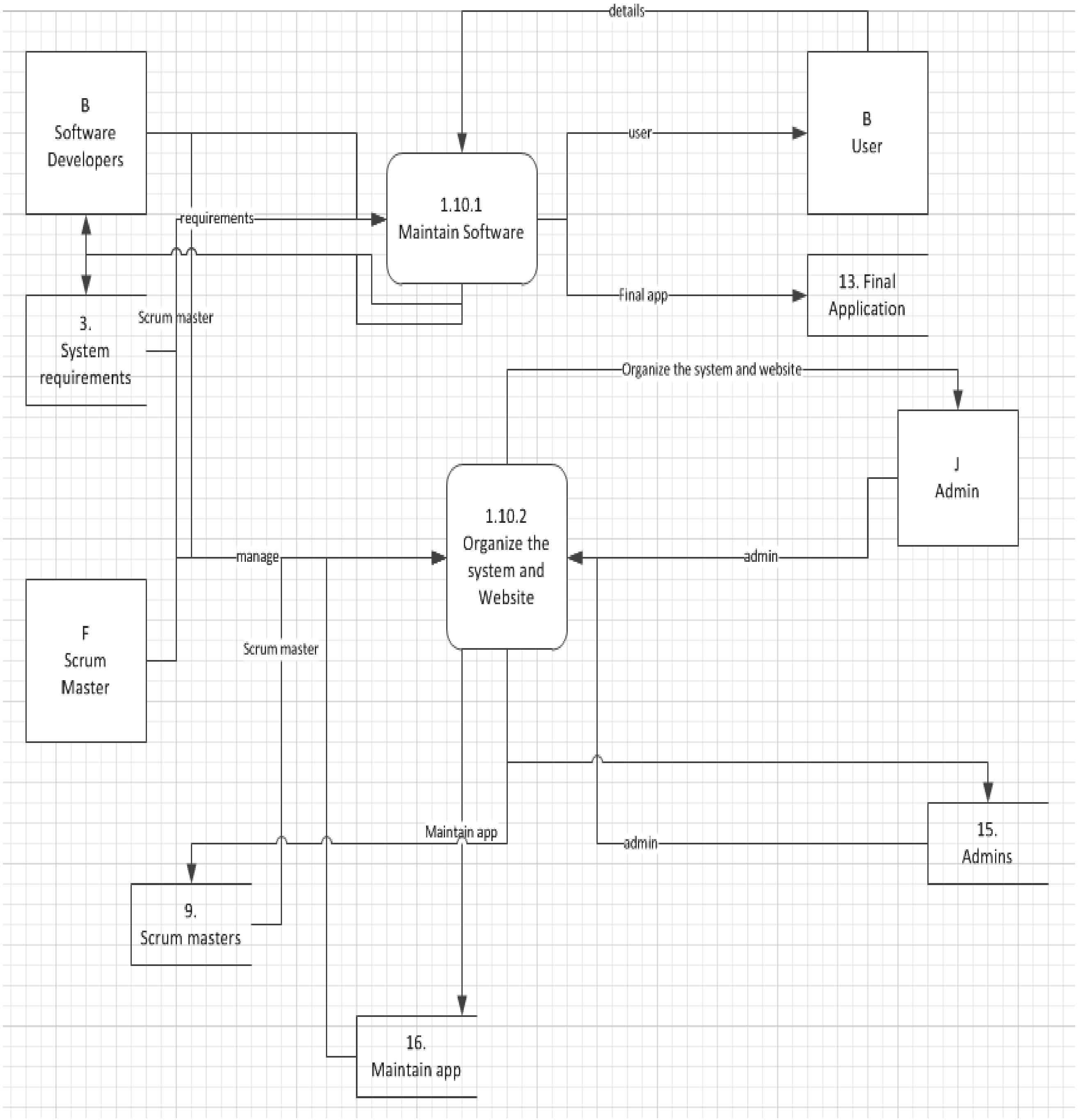
2.5.2.9 Manage the Application process

In this low level diagram the manager closes the deal with the users/clients and the scrum master can help if the manager needs it. The manager is also responsible for finding new clients/users and encourages the developers all the time.



2.5.2.10 Maintain the application

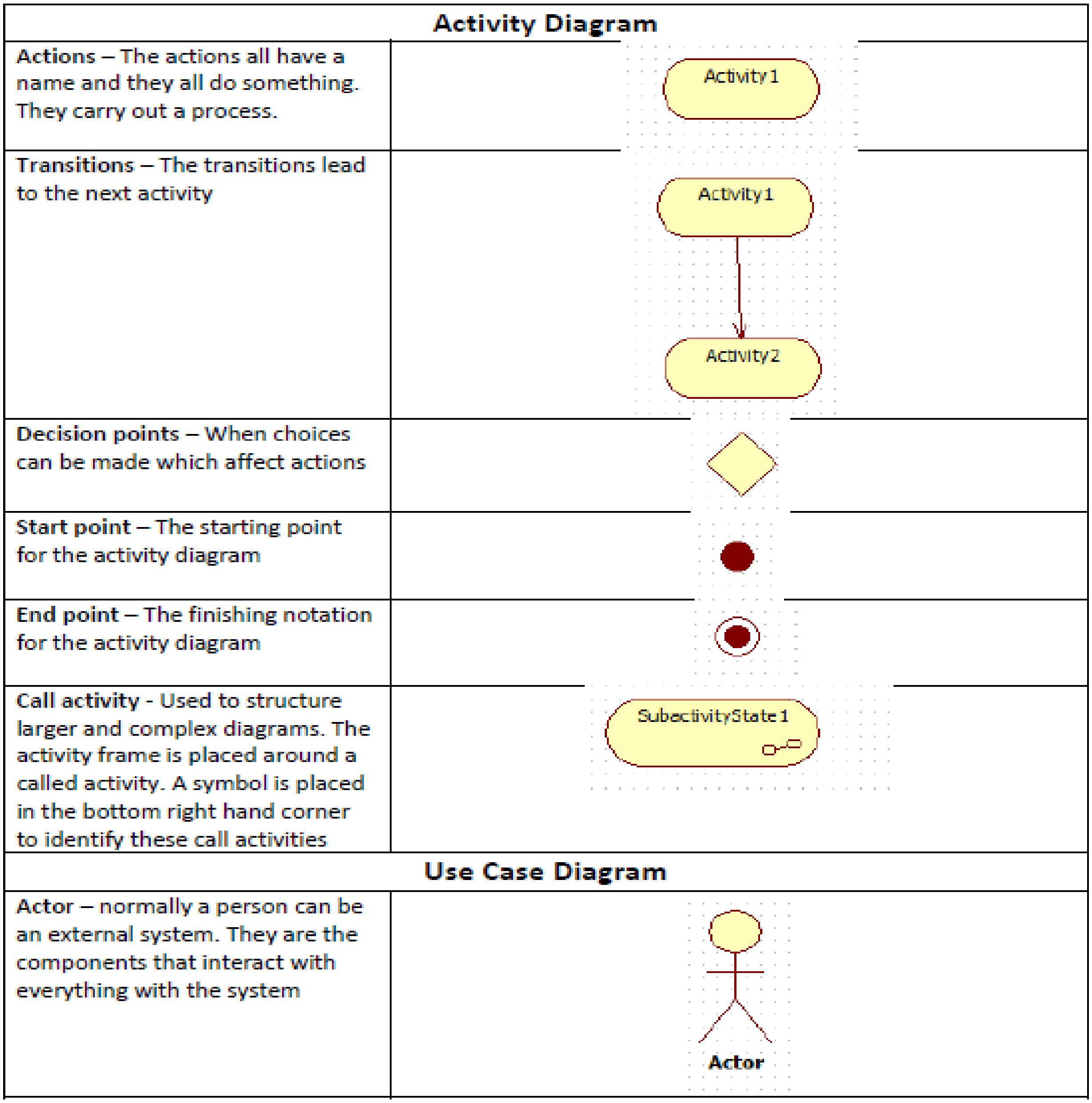
This is the last low level diagram. Here the software developers are responsible for maintaining the application. Admins are taking care of the organization of the system and the website.



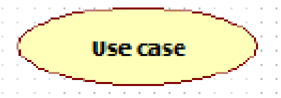
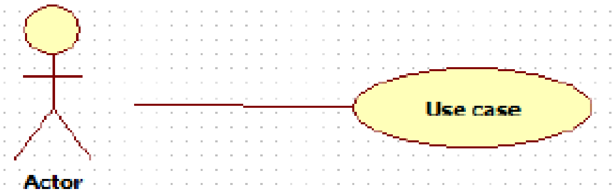
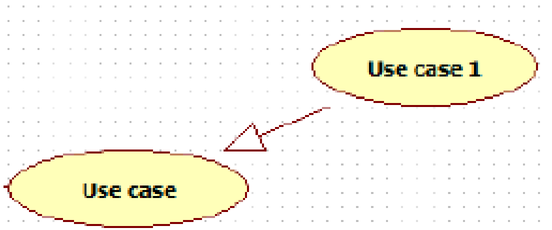
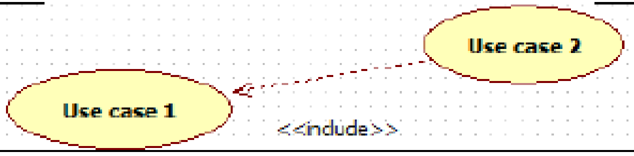
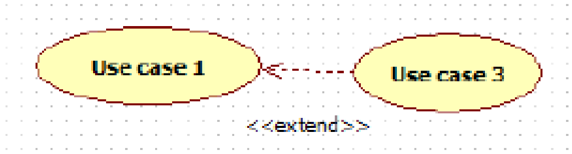
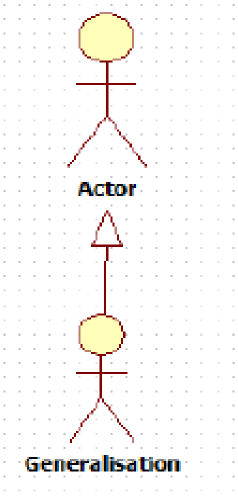
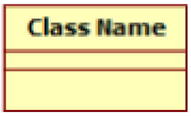
2.6 UML

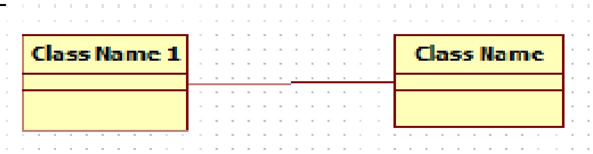
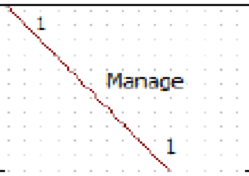
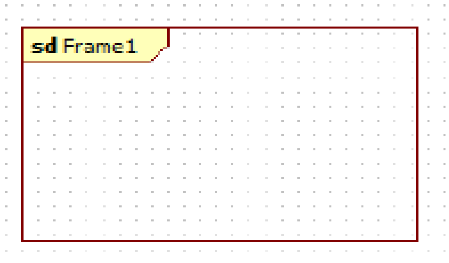
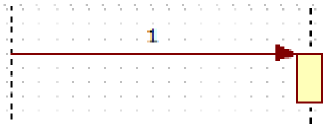
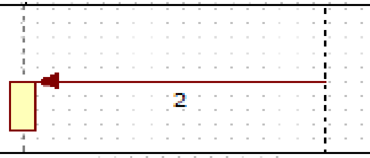
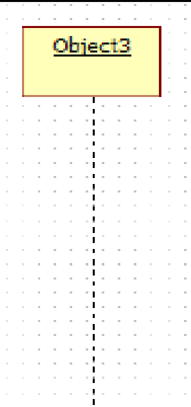
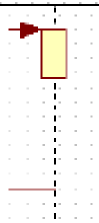
2.6.1 UML Key

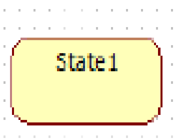
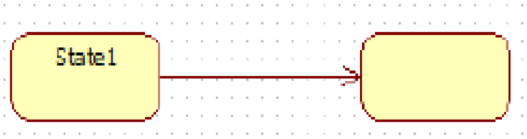
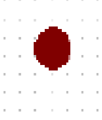
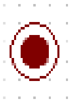
Tables below have been provided to help understand the UML notations. The tables are split in to 5 sections, one for each type of diagram. There are some notations that will be similar or the same, but they are included anyway.



Activity Diagram	
Actions – The actions all have a name and they all do something. They carry out a process.	Activity1
Transitions – The transitions lead to the next activity	Activity1 Activity2
Decision points – When choices can be made which affect actions	◇
Start point – The starting point for the activity diagram	●
End point – The finishing notation for the activity diagram	◉
Call activity - Used to structure larger and complex diagrams. The activity frame is placed around a called activity. A symbol is placed in the bottom right hand corner to identify these call activities	SubactivityState1
Use Case Diagram	
Actor – normally a person can be an external system. They are the components that interact with everything with the system	**Actor**

Use Case- A use case is a way an actor can use the system	
Association – the link between the actor and the use case. Who does what use case	
Use case generalisation – When one use case is very similar to another use case but it does something a little differently	
Include dependency – Shows that a use case 2 always includes the behaviour of the use case 1.	
Exclude dependency – Shows that use case 3's behaviour is conditionally extended by use case 1. Use case 3 will optionally use the behaviour of use case D depending on the result of some decision	
Actor generalisation – Where an actor can use all the use cases the other actor can. Shown as an arrow pointing to the actor wanting to inherit the use cases.	
Class diagram	
Class name – The classes are normally objects	

Association – The notation between classes linking them together.	
Multiplicity – It is possible for an object to be associated to a number of objects of a different class. This numerical aspect is called its multiplicity	
Sequence diagram	
Empty frame – The empty frame is a graphical boundary of a diagram.	
Incoming messages – Messages going from one object to another	
Outgoing messages – received messages going back to the objects	
Lifelines – drawing a sequence diagram, lifeline notation elements are placed across the top of the diagram. Lifelines represent either roles or object instances that participate in the sequence being modeled.	
Guards - When modeling object interactions, there will be times when a condition must be met for a message to be sent to the object. Guards are used throughout UML diagrams to control flow.	
State diagram	

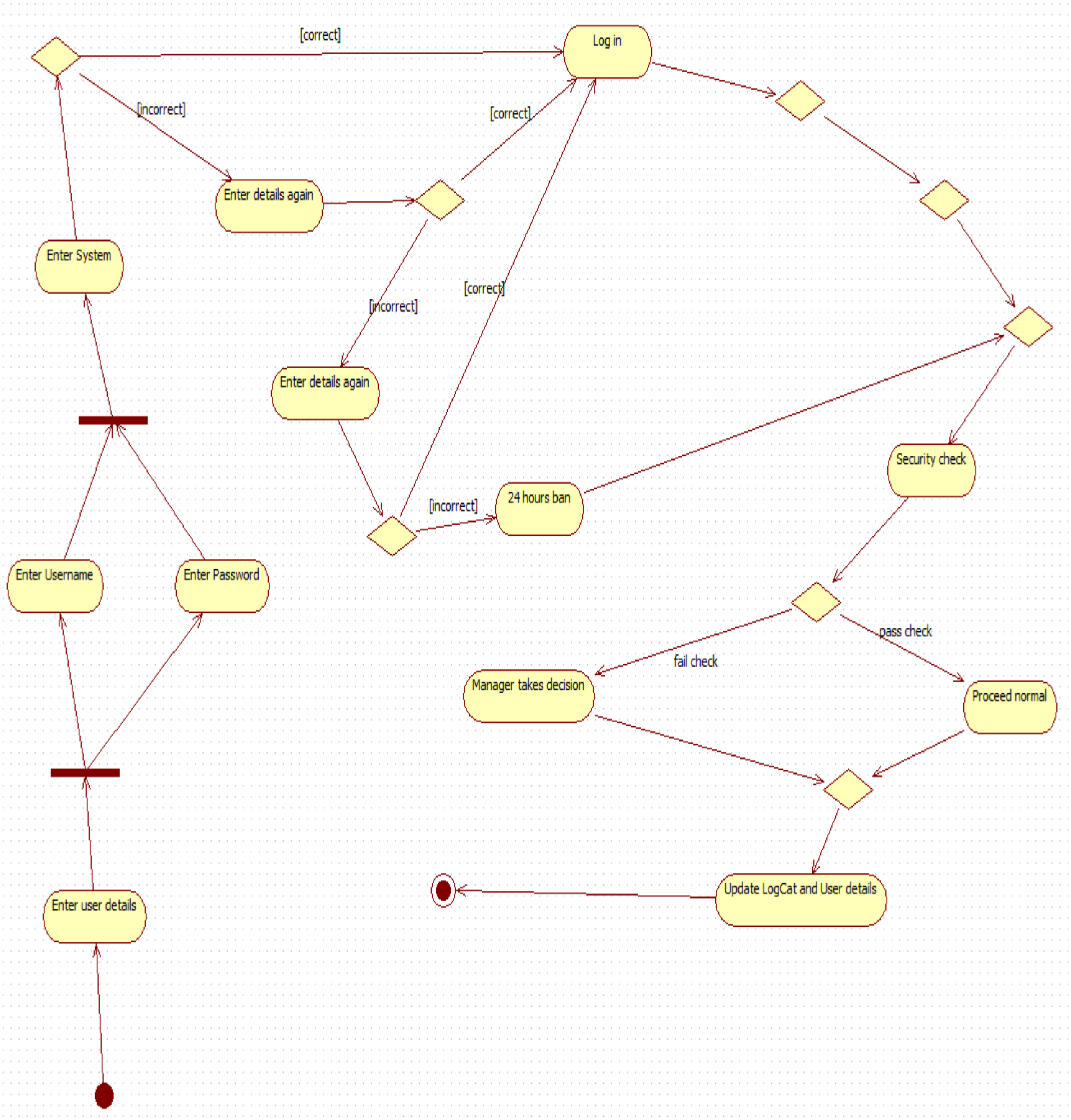
State name – represents the state of an object at an instant in time	
Transition – arrow indicating object to transition from one state to another	
Start stage – The starting point to the diagram	
Finish stage – The end stage of the diagram	

2.6.2 Activity Diagram

The first UML diagram that is going to be used is the activity diagram. The activity diagram is a graphical representation of workflows of step wise activities and actions with support for choice, iteration and concurrency. The activity diagram is used to describe the operational step by step workflows of components in a system. There have been chosen 5 components to describe using the activity diagram due to the choices to be made by the user/client in order for them to be completed. The other processes were straight forward and were not worth going in to details.

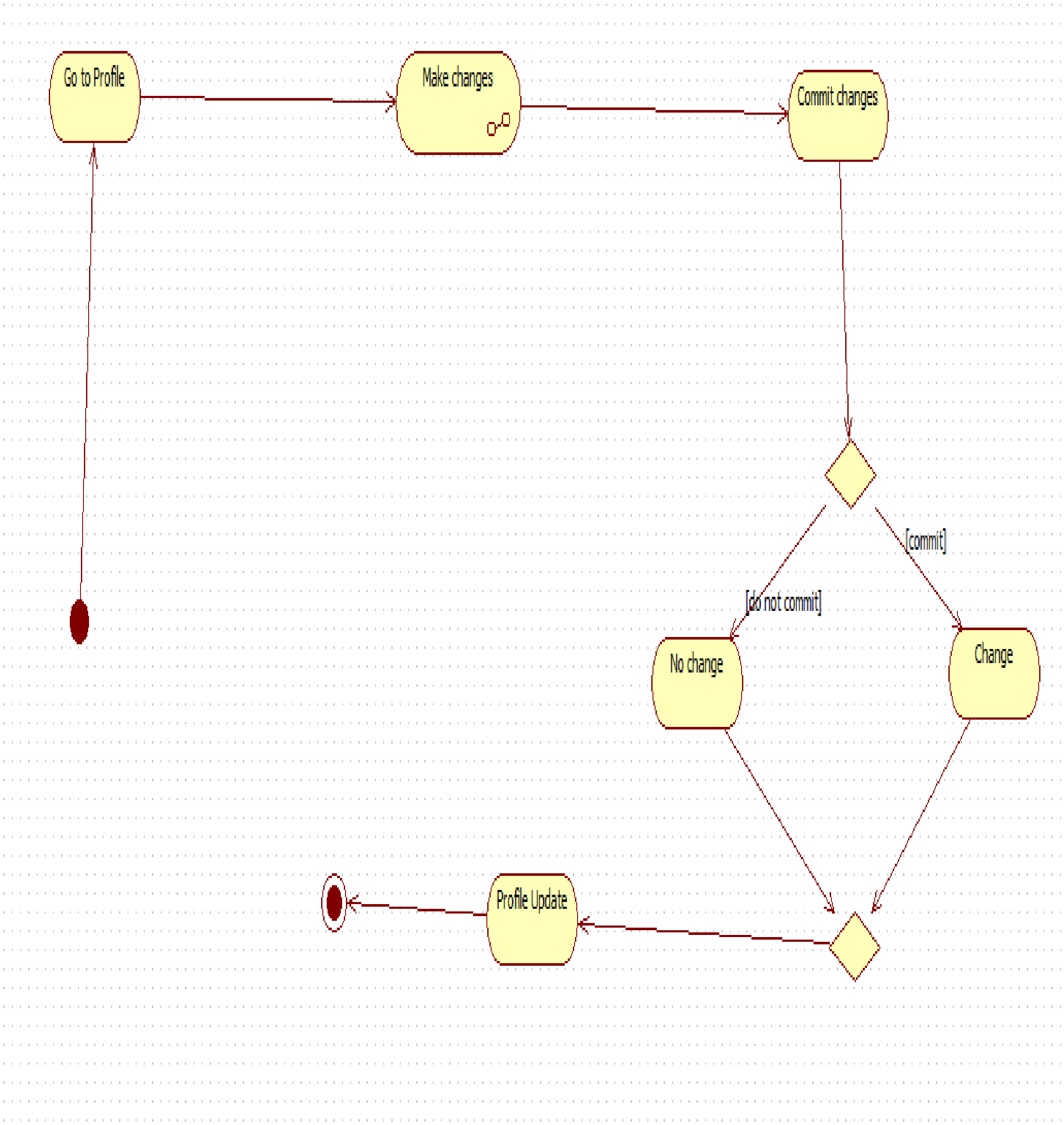
2.6.2.1 Log in

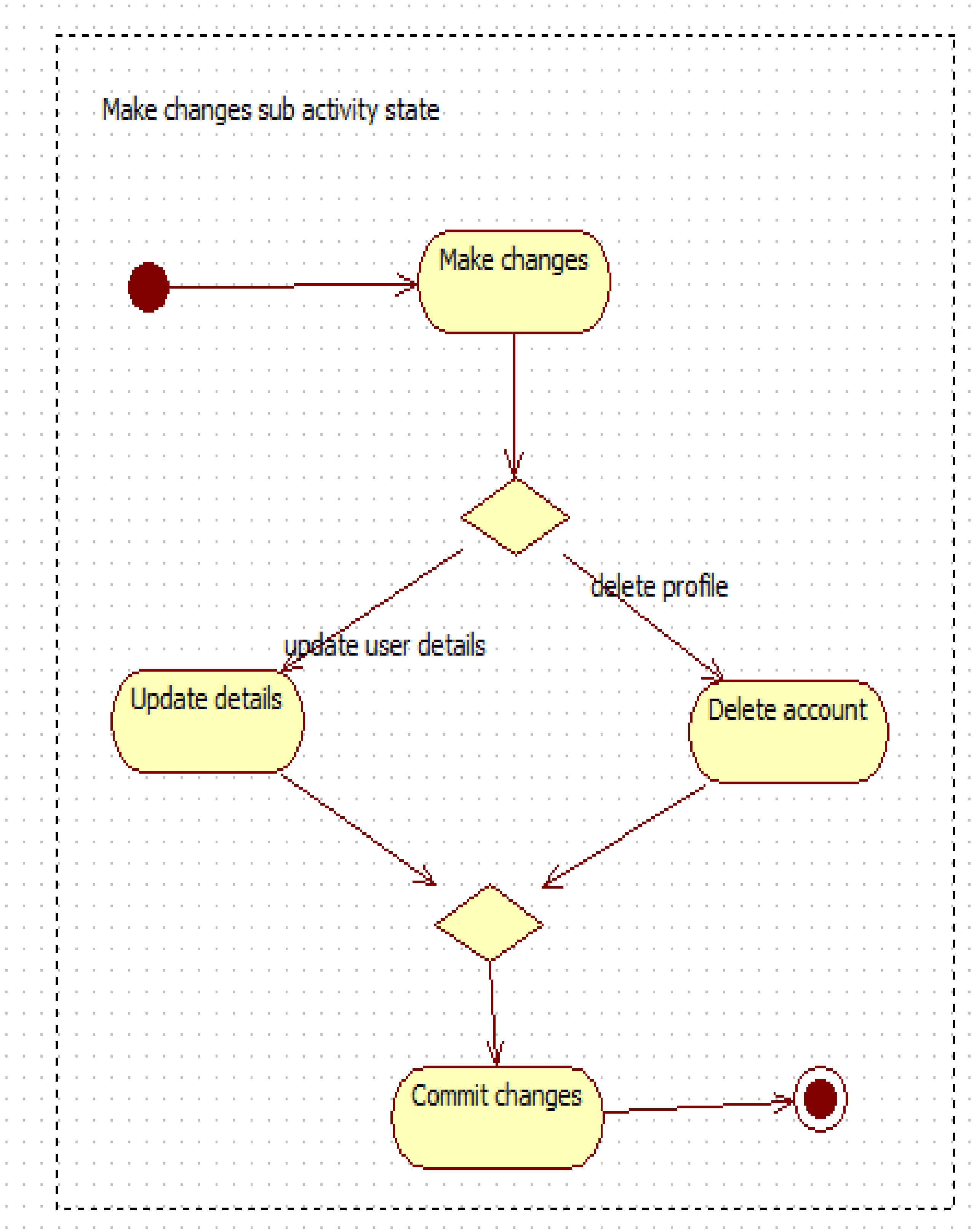
The log in process in the activity diagrams below only shows the username and password entry. Once entered, the activity diagram goes in to a series of decision points. If entered correctly the user gets logged in, if not the decision points continue until the user has entered incorrectly 3 times, a ban is given and then a security check is made and the diagram ends with the update of the log cat and user details.



2.6.2.2 Profile Update

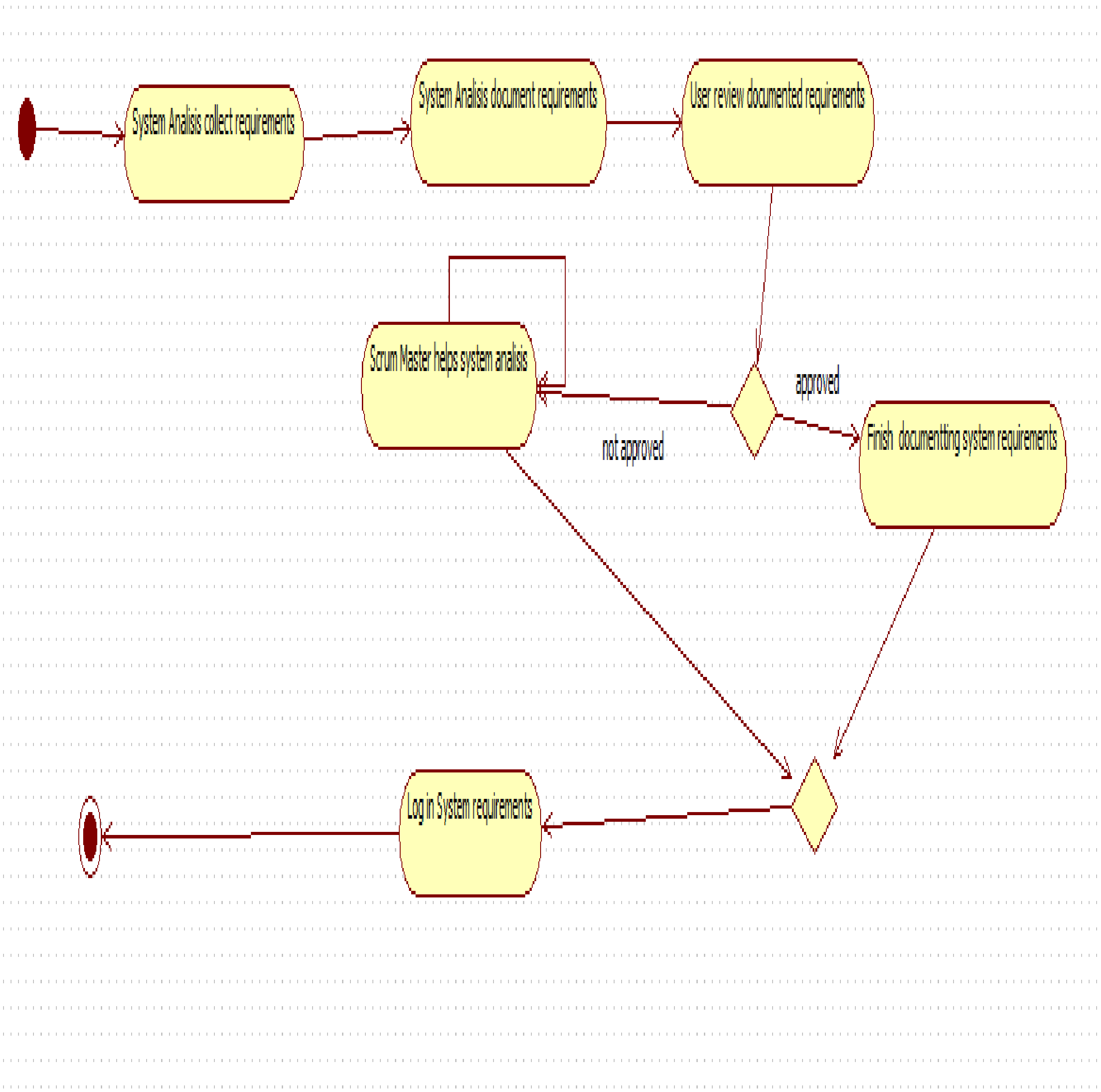
The profile update in the activity diagrams below show the changes that a user can make. The user can update his personal details and then commit the changes or he can delete his profile. This is shown into the Make changes sub activity diagram.




Make changes sub activity state
Make changes
delete profile
update user details
Update details
Delete account
Commit changes

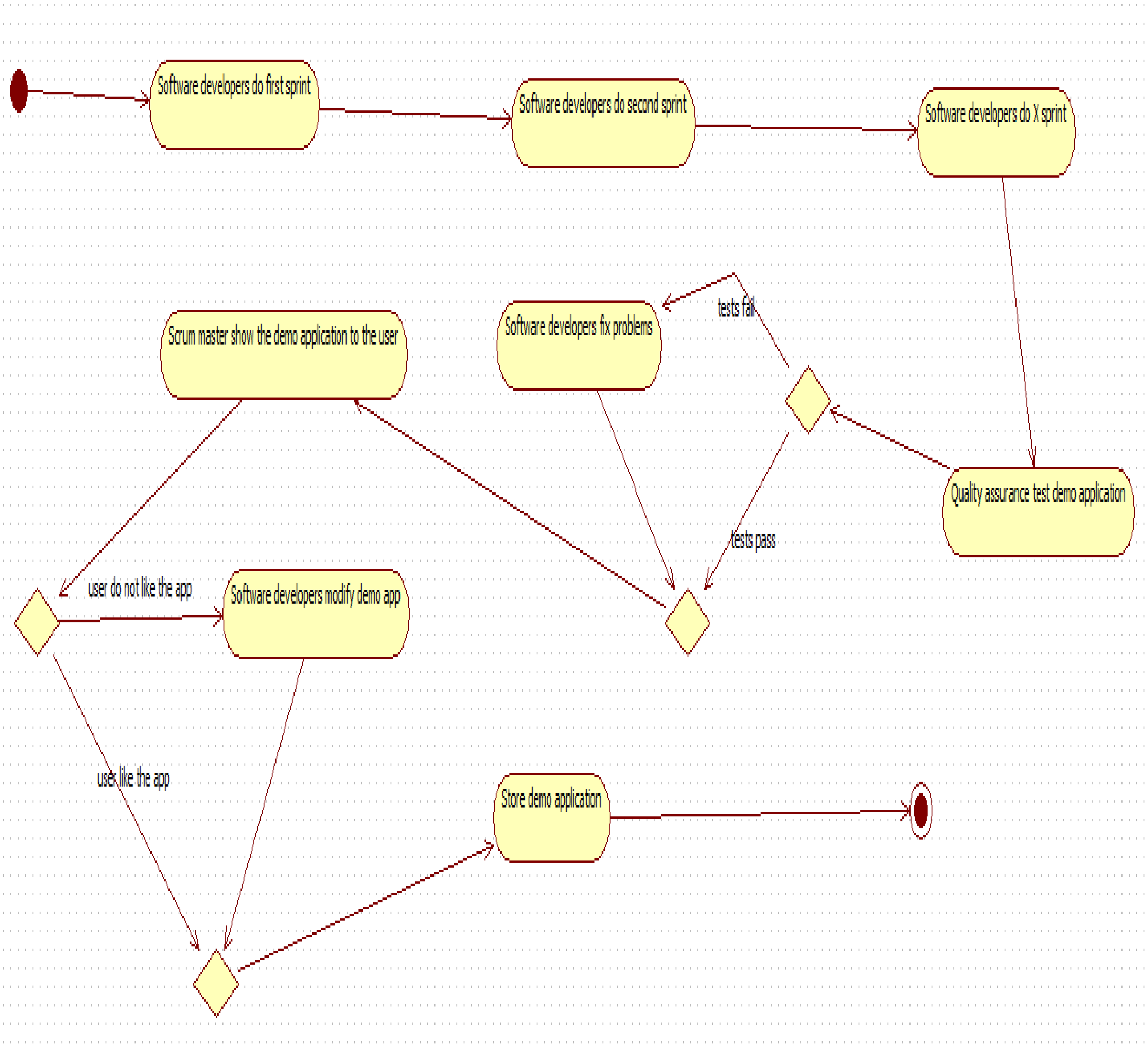
2.6.2.3 System Requirements

The system requirements activity diagram shows below how the system requirements are collected from the user/client. If the client is not happy with the documentation then the scrum master is going to give a hand to the system analysis t complete them.


System Analisis collect requirements
System Analisis document requirements
User review documented requirements
Scrum Master helps system analisis
approved
not approved
Finish documentting system requirements
Log in System requirements

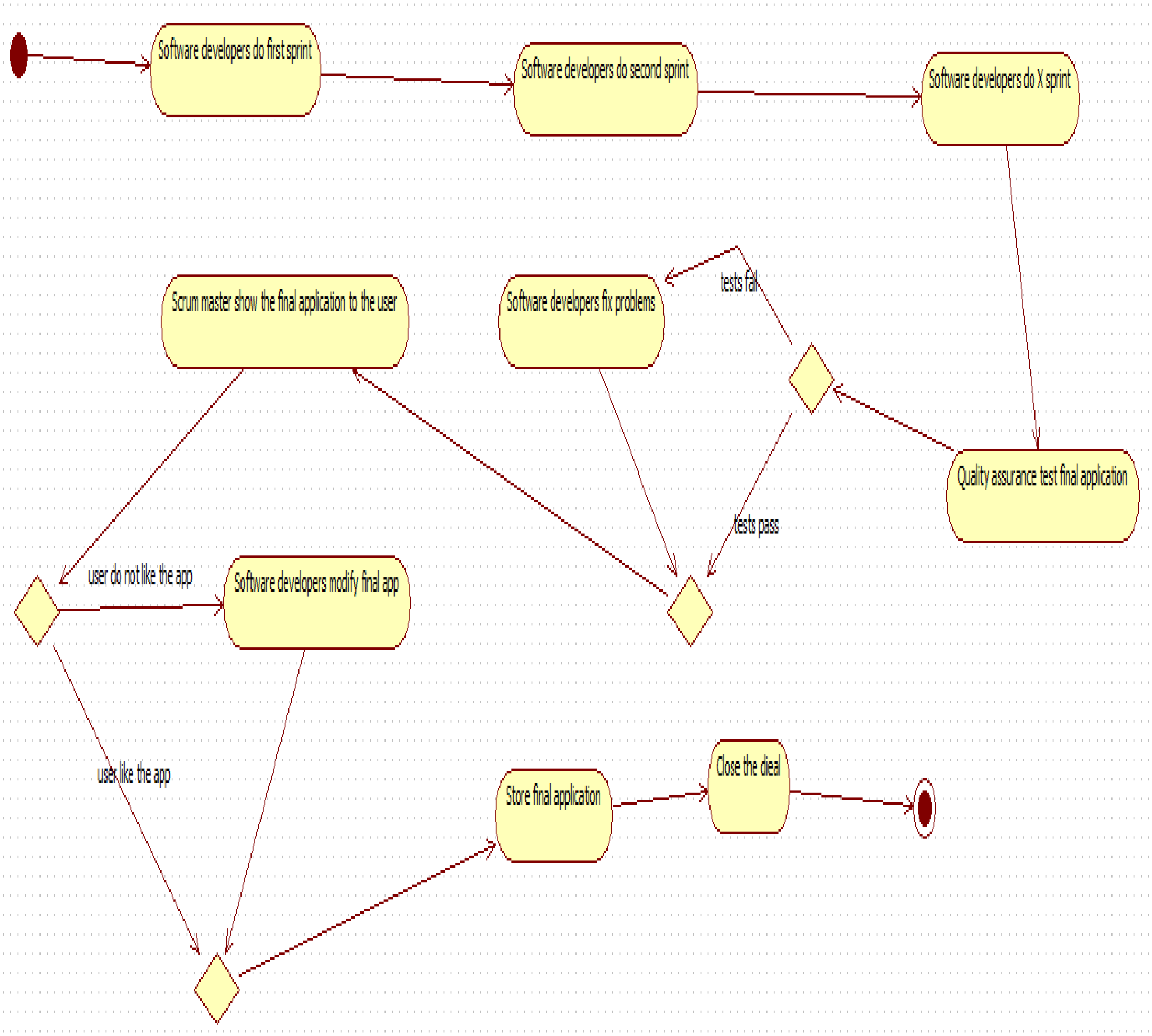
2.6.2.4 Demo Application

The demo application activity diagram below shows how a sample demo application is being developed. First the Software developers do N amount of sprints which usually are between 2 and 6 and each sprint is between 1 week and 1 month. Then the application is being tested and if not any major issues found the demo application is being shown to the user by the scrum master.


Software developers do first sprint
Software developers do second sprint
Software developers do X sprint
tests fail
Software developers fix problems
Scrum master show the demo application to the user
Quality assurance test demo application
tests pass
user do not like the app
Software developers modify demo app
user like the app
Store demo application

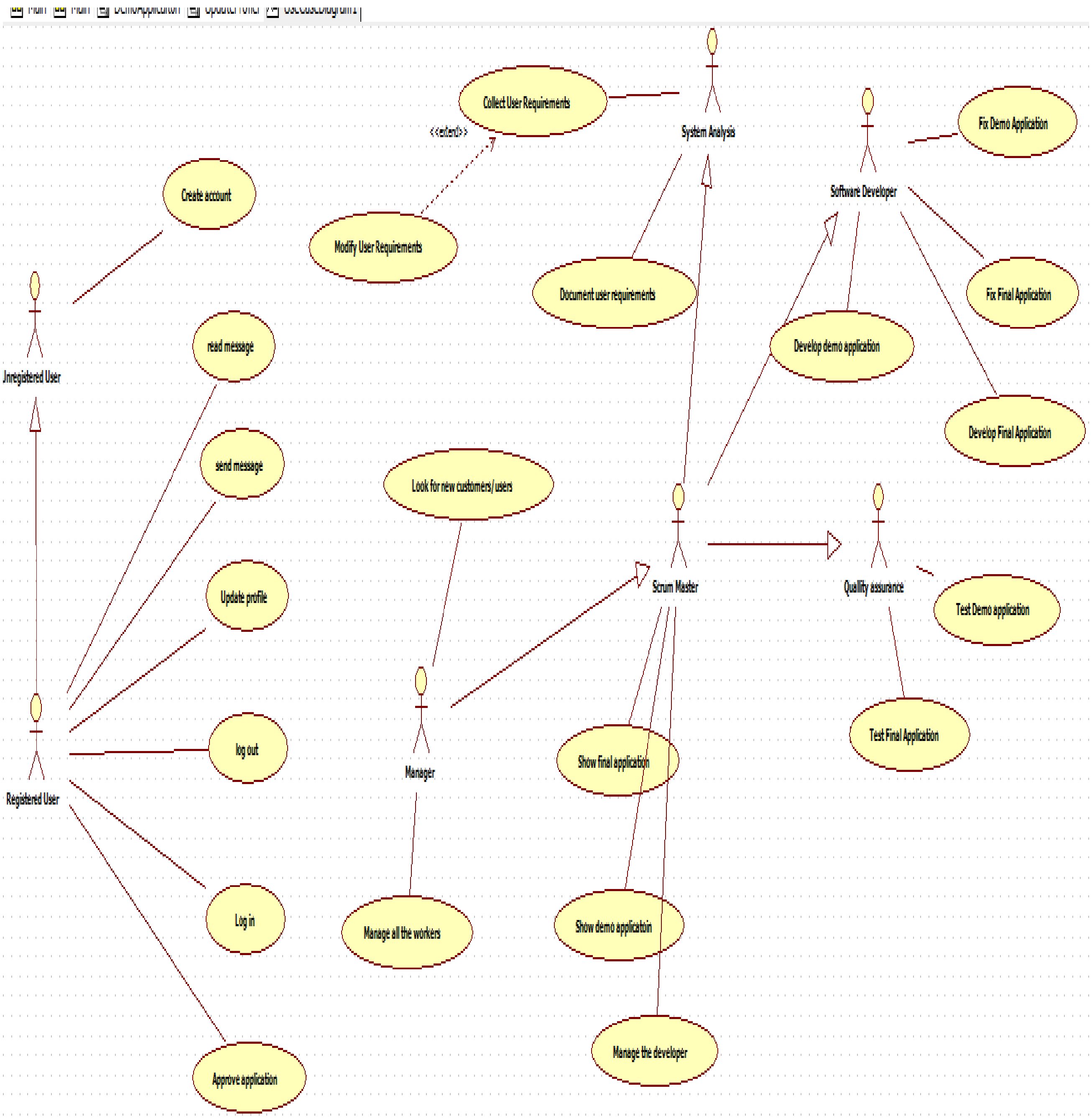
2.6.2.5 Final Application

The final application activity diagram below shows how a sample final application is being developed. First the Software developers do N amount of sprints which usually are between 2 and 6 and each sprint is between 1 week and 1 month. Then the application is being tested and if not any major issues found the final application is being shown to the user by the scrum master and the deal is finalized.


Software developers do first sprint
Software developers do second sprint
Software developers do X sprint
Scrum master show the final application to the user
Software developers fix problems
tests fail
Quality assurance test final application
tests pass
user do not like the app
Software developers modify final app
user like the app
Store final application
Close the dieal

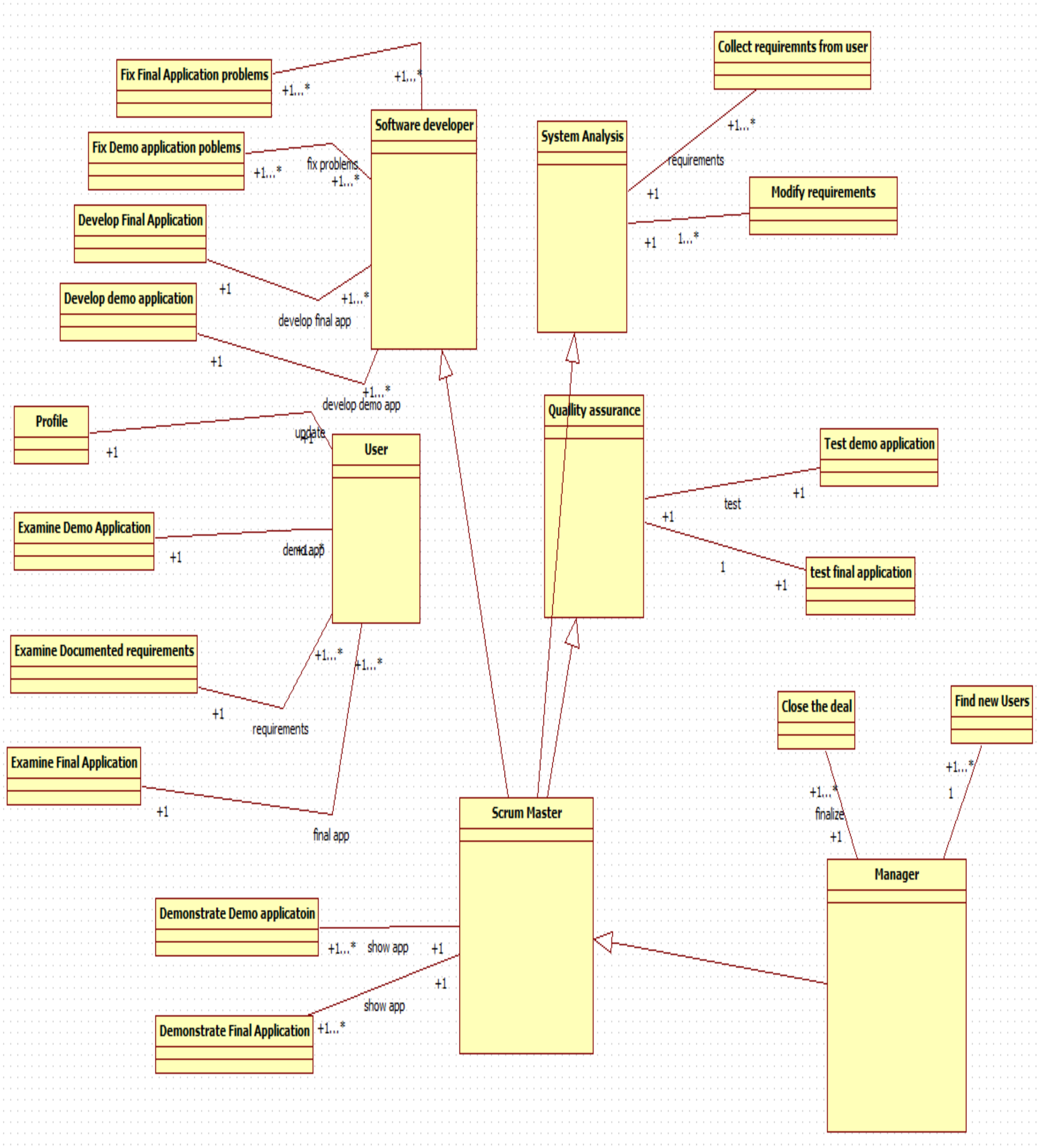
2.6.3 Use Case Diagram

The use case diagram is a list of steps defining interactions between an actor and a system to achieve a goal. The actor can be either human or an external system the use case diagram below shows all actors interacting with the system. There are 7 actors interacting with the system. The lower level processes weren't included in the use case diagram due to their little importance.



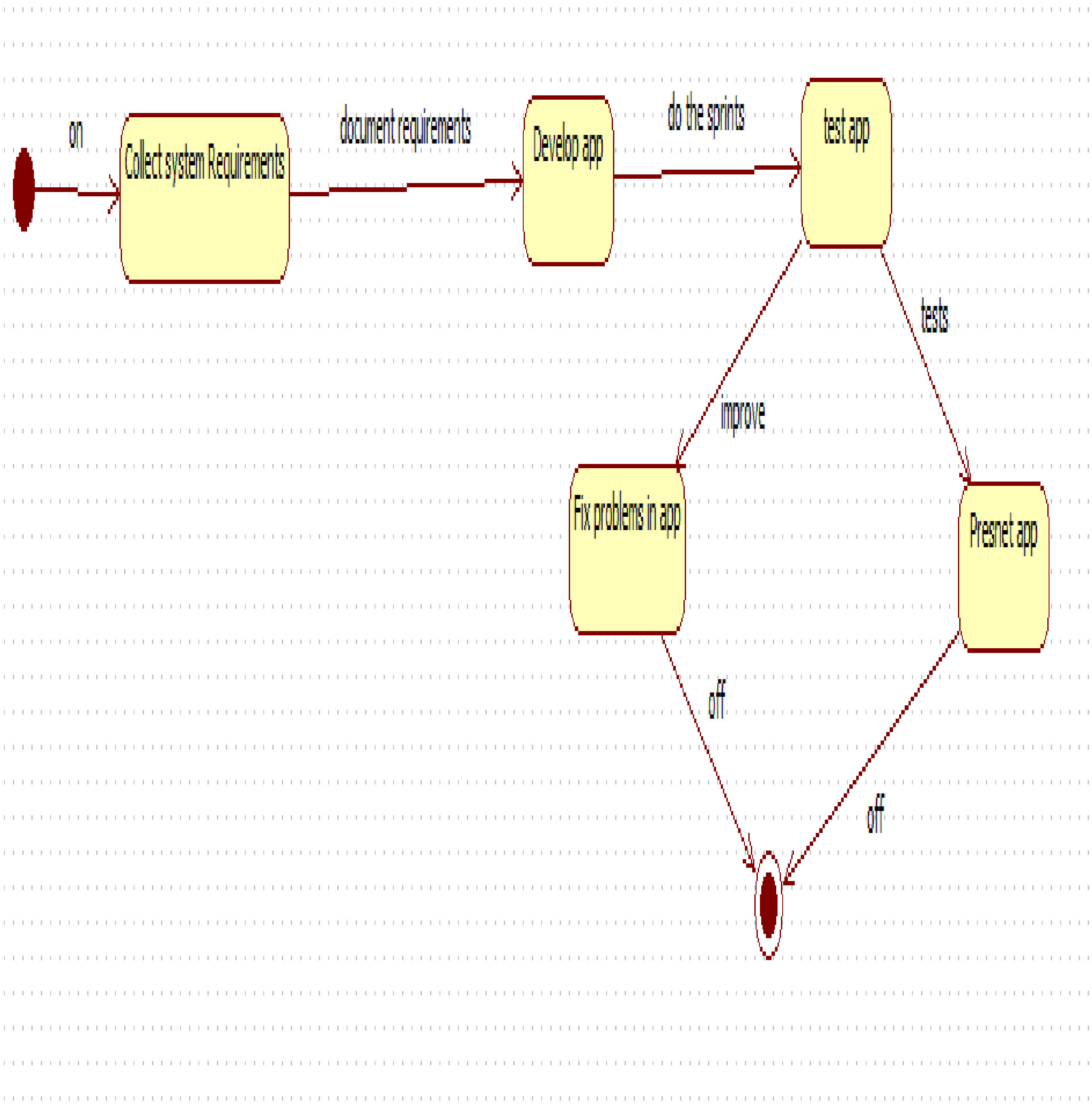
2.6.4 Class diagram

The class diagram shows the functionality of the system (what the actors can do). The use case specifications capture interaction processes between user and the system. The systems classes have to identify, relationships and multiplicities. Once the objects in the system have been identified they have to be grouped in to a class diagram.



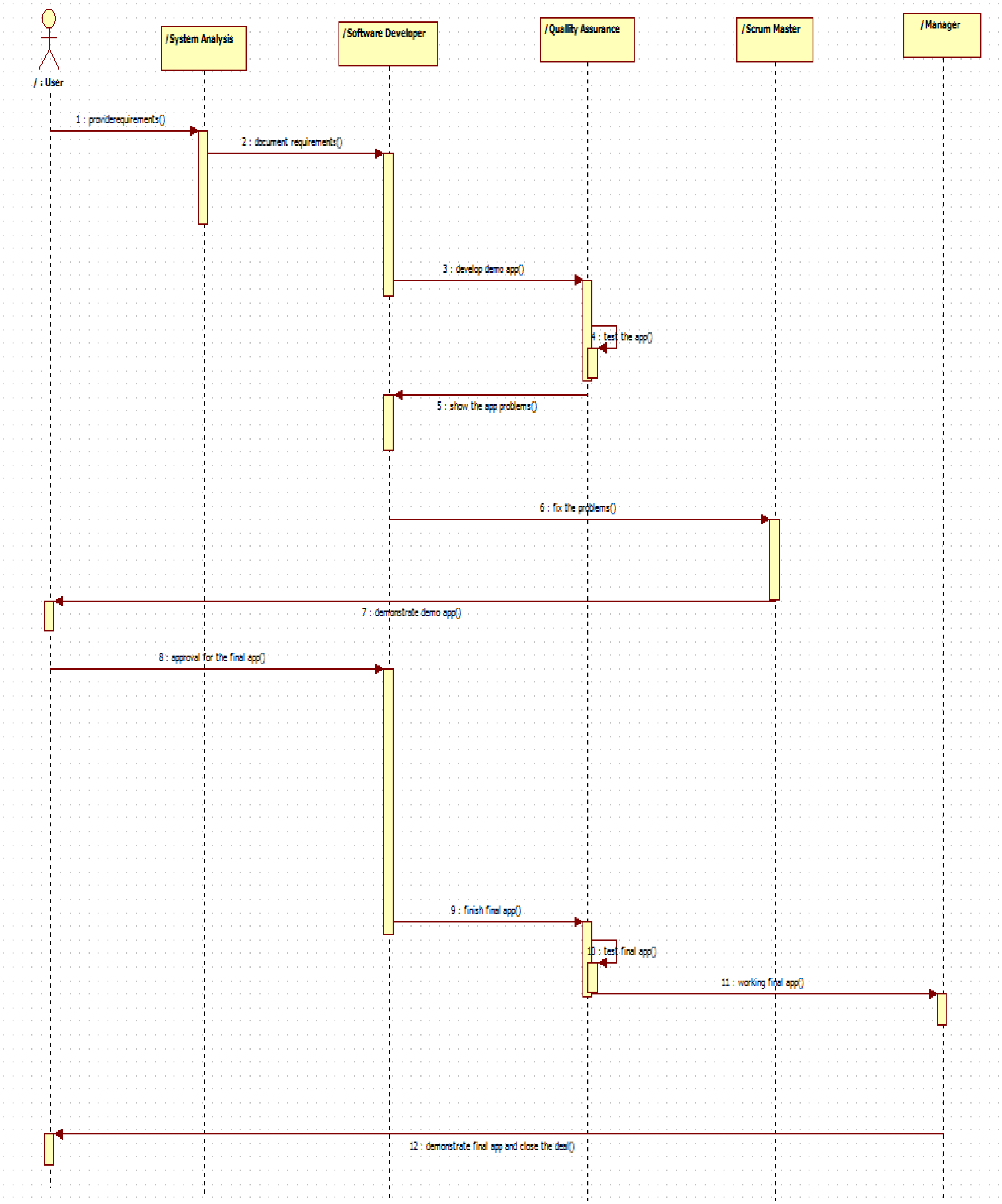
2.6.5 State Diagram

The state diagrams is to demonstrate the behavior of an object through many use cases of the system this diagram is showing the behavior of the application development classes from the class diagram.



2.6.6 Sequence diagram

Sequence diagram is a kind of diagram that shows how processes operate with one another and in what order.



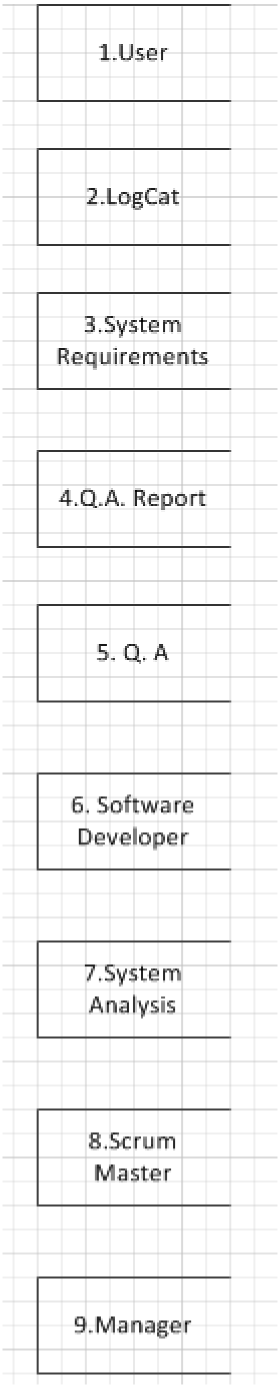
3. Design Specification

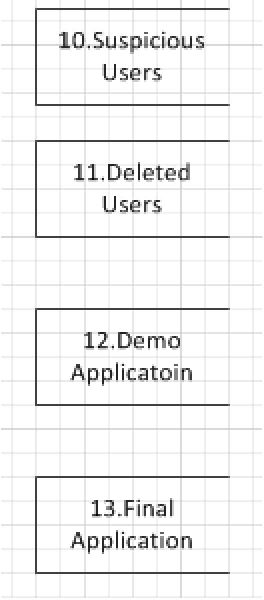
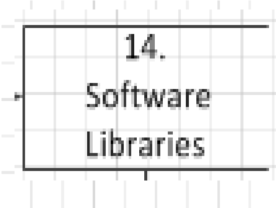
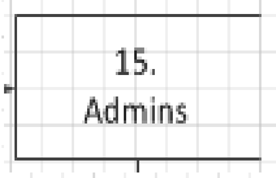
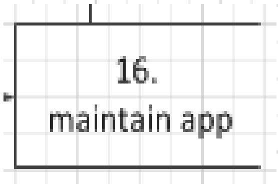
3.1Data dictionary

Each data store requires a data description compromising of its data structures and elements. The data structures are a collection of elements and structures. The structures are broke down in to elements that cannot be broke down any further. Each element will be described (that has been broken down from the structures in that data store) with their data type and the format they will have. A few structures and elements have been repeated in each data stores as they were necessary for better comprehension. Without the repetition of some of the structures, the data store would not make much sense to most of the people. The data structures link together to make the data store meaningful. Naming the elements again could be avoided and it has been taken in to account. Repetition is not a good quality in report writing but when it is needed, it was put it in for extra clarity.

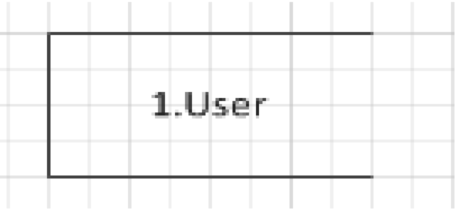
3.1.1 Data stores – structures and elements

1. **User**
2. **LogCat**
3. **System Requirements**
4. **Q.A Report**
5. **Q.A**
6. **Software Developer**
7. **System Analysis**
8. **Scrum Master**
9. **Manager**



10. Suspicious Users	 10.Suspicious Users 11.Deleted Users 12.Demo Applicatoin 13.Final Application
11. Deleted Users	
12. Demo Application	
13. Final Application	
14. Software Libraries	
15. Admins	
16. Maintain app	

3.1.1.1 Data Store 1 – User



Data Structures:

USER ID = @UserIdNumber+Username+Password+EmailAddress+CreationDate

USER = Title+Forename+Surname+DOB+Gender+ADDRESS

ADDRESS = HouseNumber+AddressLine1+ (AddressLine2) +City+County+PostCode+ (ZIPCode) +Country

CONTACT DETAILS = HomeTelephoneNumber+ (MobileTelephoneNumber) +EmailAddress

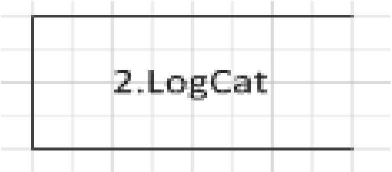
PROFILE = AboutMe+ (Industry)+ (Software)

Data Elements

Element	Data	Format/Description
USERID		
UserIdNumber	7{Integer}7	####### each # is [0…9]
Username		1{Char}20
Password		1{Char}20
EmailAddress		1{Char}20
CreationDate	date	##-##-#### each # is [0…9]
USER		
Title	1{char}4	Usually from [Mr|Mrs|Master|Miss]
Forename		1{char}20
Surname		1{char}20
DOB	date	##-##-#### each # is [0…9]
Gender	Boolean	List 0 = Male 1 = Female
ADDRESS		
HouseNumber	1{char}3	Usually numeric but can be subset [0…9|A…Z|a…z]
AddressLine1		1{char}30
AddressLine2		1{char}30
City		1{char}30
County		1{char}30
Postcode	7{char}7	Format $$# %$$ where$= [A…Z] #=[1..99] and %=[0..9]

ZIPCode	7{char}7	Format $$# %$$ where$= [A...Z] #=[1..99] and %=[0..9]
Country	1{char} 25	
CONTACT DETAILS		
HomeTelephoneNumber	integer	Varies due to phone length in different countries
MobileTelephoneNumber	integer	
EmailAddress	1{char}20	
PROFILE		
AboutMe	1{char}MAX	
Industry	[A...Z|a...z]	
Software	1{char}20	

3.1.1.2 Data Store LogCat



Data structures:

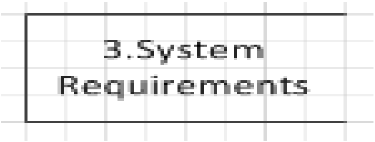
LogCat = @UserID+UserEmailAddress+LastLogInDate+AttemptsMade+UsernameRemindersMade+PasswordRemindersMade

PASSWORD CHANGE = LastPasswordChangeDate+OldPassword+NewPassword

Data Elements

Element	Data	Format/Description
LogCat		
UserID	7{integer}7	####### each # is [0...9]
UserEmailAddress	1{char}20	
LastLogInDate	date	##-##-#### each # is [0...9]
AttemptsMade	1{integer}MAX	
UsernameRemindersMade	1{integer}MAX	
PasswordRemindersMade	1{integer}MAX	
PASSWORD CHANGE		
LastPasswordChangeDate	date	##-##-#### each # is [0...9]
OldPassword	1{char}20	
NewPassword	1{char}20	

3.1.1.3 Data Store System Requirements



Data structures:

User requirements = @UserIdNumber+Username+Password+EmailAddress+CreationDate +System Requirements

System Analysis = Title+Forename+Surname+DOB+Gender + Documented System requirements +ADDRESS

ADDRESS = HouseNumber+AddressLine1+ (AddressLine2) +City+County+PostCode+ (ZIPCode) +Country

Data Elements

Element	Data	Format/Description
USER Requirements		
UserIdNumber	7{Integer}7	####### each # is [0…9**]**
Username		1{Char}20
Password		1{Char}20
EmailAddress		1{Char}20
CreationDate	date	##-##-#### each # is [0…9]
System Requirements(Raw)	Date	Pdf file/ paper
System Analysis		
Title	1{char}4	Usually from [Mr|Mrs|Master|Miss]
Forename		1{char}20
Surname		1{char}20
DOB	date	##-##-#### each # is [0…9]
Gender	Boolean	List 0 = Male 1 = Female
Documented System requirements (official)	Date	Pdf file
ADDRESS		
HouseNumber	1{char}3	Usually numeric but can be subset [0...9|A...Z|a...z]
AddressLine1		1{char}30
AddressLine2		1{char}30
City		1{char}30
County		1{char}30
Postcode	7{char}7	Format $$# %$$ where$= [A...Z] #=[1..99] and %=[0..9]

3.1.1.4 Data Store Q.A Report

Data structures:

Software Expertise = @ TypeOfSoftware1 + TypeOfSoftware2 + TypeOfSoftwareN

Quality Assurance = Title+Forename+Surname+DOB+Gender + Documented System requirements +ADDRESS

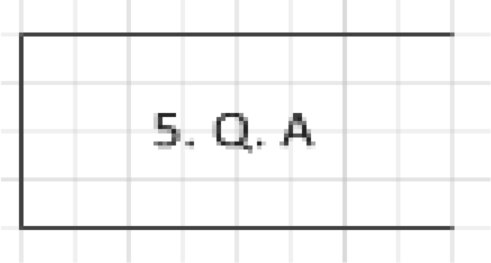
ADDRESS = HouseNumber+AddressLine1+ (AddressLine2) +City+County+PostCode+ (ZIPCode) +Country

Quality Assurance result

Data Elements

Element	Data	Format/Description
Software Expertise		
TypeOfSoftware1	1{Char}20	
TypeOfSoftware2	1{Char}20	
TypeOfSoftwareN	1{Char}20	
Quality Assurance		
Title	1{char}4	Usually from [Mr|Mrs|Master|Miss]
Forename	1{char}20	
Surname	1{char}20	
DOB	date	##-##-#### each # is [0…9]
Gender	Boolean	List 0 = Male 1 = Female
ADDRESS		
HouseNumber	1{char}3	Usually numeric but can be subset [0…9|A…Z|a…z]
AddressLine1	1{char}30	
AddressLine2	1{char}30	
City	1{char}30	
County	1{char}30	
Postcode	7{char}7	Format $$# %$$ where$= [A…Z] #=[1..99] and %=[0..9]
Quality Assurance result		
Results	Boolean	0 – Failed, 1 Passed

3.1.1.5 Data Store Q.A



Data structures:

Area of Expertise = @Web development + Wed design + Mobile development + Microsoft Development + Java Developement

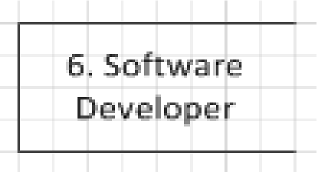
Q.A = Title+Forename+Surname+DOB+Gender + Documented System requirements +ADDRESS

ADDRESS = HouseNumber+AddressLine1+ (AddressLine2)
+City+County+PostCode+ (ZIPCode) +Country

Data Elements

Element	Data	Format/Description
Area Expertise		
Web development	Boolean	0-false, 1-true
Wed design	Boolean	0-false, 1-true
Mobile development	Boolean	0-false, 1-true
Microsoft development	Boolean	0-false, 1-true
Java development	Boolean	0-false, 1-true
Q.A		
Title	1{char}4	Usually from [Mr|Mrs|Master|Miss]
Forename	1{char}20	
Surname	1{char}20	
DOB	date	##-##-#### each # is [0...9]
Gender	Boolean	List 0 = Male 1 = Female
ADDRESS		
HouseNumber	1{char}3	Usually numeric but can be subset [0...9|A...Z|a...z]
AddressLine1	1{char}30	
AddressLine2	1{char}30	
City	1{char}30	
County	1{char}30	
Postcode	7{char}7	Format $$# %$$ where$= [A...Z] #=[1..99] and %=[0..9]

3.1.1.6 Data Store Software Developer



Data structures:

Area of Expertise = @Web development + Wed design + Mobile development + Microsoft Development + Java Developement

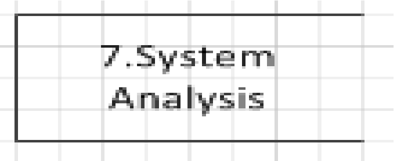
Software Developer = Title+Forename+Surname+DOB+Gender + Documented System requirements +ADDRESS

ADDRESS = HouseNumber+AddressLine1+ (AddressLine2) +City+County+PostCode+ (ZIPCode) +Country

Data Elements

Element	Data	Format/Description
Area Expertise		
Web development	Boolean	0-false, 1-true
Wed design	Boolean	0-false, 1-true
Mobile development	Boolean	0-false, 1-true
Microsoft development	Boolean	0-false, 1-true
Java development	Boolean	0-false, 1-true
Software Developer		
Title	1{char}4	Usually from [Mr|Mrs|Master|Miss]
Forename	1{char}20	
Surname	1{char}20	
DOB	date	##-##-#### each # is [0…9]
Gender	Boolean	List 0 = Male 1 = Female
ADDRESS		
HouseNumber	1{char}3	Usually numeric but can be subset [0...9|A...Z|a...z]
AddressLine1	1{char}30	
AddressLine2	1{char}30	
City	1{char}30	
County	1{char}30	
Postcode	7{char}7	Format $$# %$$ where$= [A...Z] #=[1..99] and %=[0..9]

3.1.1.7 Data Store System Analysis



Data structures:

Area of Expertise = @Web development + Wed design + Mobile development + Microsoft Development + Java Developement

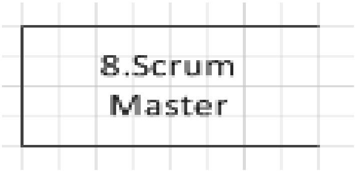
System Analysis = Title+Forename+Surname+DOB+Gender + Documented System requirements +ADDRESS

ADDRESS = HouseNumber+AddressLine1+ (AddressLine2) +City+County+PostCode+ (ZIPCode) +Country

Data Elements

Element	Data	Format/Description
Area Expertise		
Web development	Boolean	0-false, 1-true
Wed design	Boolean	0-false, 1-true
Mobile development	Boolean	0-false, 1-true
Microsoft development	Boolean	0-false, 1-true
Java development	Boolean	0-false, 1-true
System Analysis		
Title	1{char}4	Usually from [Mr|Mrs|Master|Miss]
Forename	1{char}20	
Surname	1{char}20	
DOB	date	##-##-#### each # is [0…9]
Gender	Boolean	List 0 = Male 1 = Female
ADDRESS		
HouseNumber	1{char}3	Usually numeric but can be subset [0…9|A…Z|a…z]
AddressLine1	1{char}30	
AddressLine2	1{char}30	
City	1{char}30	
County	1{char}30	
Postcode	7{char}7	Format $$# %$$ where$= [A…Z] #=[1..99] and %=[0..9]

3.1.1.8 Data Store Scrum Master



Data structures:

Area of Expertise = @Web development + Wed design + Mobile development + Microsoft Development + Java Developement

Scrum Master = Title+Forename+Surname+DOB+Gender + Documented System requirements +ADDRESS

ADDRESS = HouseNumber+AddressLine1+ (AddressLine2) +City+County+PostCode+ (ZIPCode) +Country

Data Elements

Element	Data	Format/Description
Area Expertise		
Web development	Boolean	0-false, 1-true
Wed design	Boolean	0-false, 1-true
Mobile development	Boolean	0-false, 1-true
Microsoft development	Boolean	0-false, 1-true
Java development	Boolean	0-false, 1-true
Scrum Master		
Title	1{char}4	Usually from [Mr|Mrs|Master|Miss]
Forename	1{char}20	
Surname	1{char}20	
DOB	date	##-##-#### each # is [0…9]
Gender	Boolean	List 0 = Male 1 = Female
ADDRESS		
HouseNumber	1{char}3	Usually numeric but can be subset [0…9|A…Z|a…z]
AddressLine1	1{char}30	
AddressLine2	1{char}30	
City	1{char}30	
County	1{char}30	
Postcode	7{char}7	Format $$# %$$ where$= [A…Z] #=[1..99] and %=[0..9]

3.1.1.9 Data Store Manager

Data structures:

Area of Expertise = @Web development + Wed design + Mobile development + Microsoft Development + Java Developement

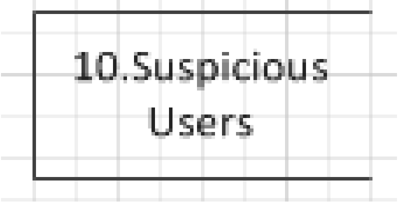
***Manager* = Title**+Forename+Surname+DOB+Gender + Documented System requirements +ADDRESS

ADDRESS = HouseNumber+AddressLine1+ (AddressLine2) +City+County+PostCode+ (ZIPCode) +Country

Data Elements

Element	Data	Format/Description
Area Expertise		
Web development	Boolean	0-false, 1-true
Wed design	Boolean	0-false, 1-true
Mobile development	Boolean	0-false, 1-true
Microsoft development	Boolean	0-false, 1-true
Java development	Boolean	0-false, 1-true
Manager		
Title	1{char}4	Usually from [Mr|Mrs|Master|Miss]
Forename	1{char}20	
Surname	1{char}20	
DOB	date	##-##-#### each # is [0...9]
Gender	Boolean	List 0 = Male 1 = Female
ADDRESS		
HouseNumber	1{char}3	Usually numeric but can be subset [0...9|A...Z|a...z]
AddressLine1	1{char}30	
AddressLine2	1{char}30	
City	1{char}30	
County	1{char}30	
Postcode	7{char}7	Format $$# %$$ where$= [A...Z] #=[1..99] and %=[0..9]

3.1.1.10 Data Store Suspicious Users



Data Structures:

Suspicious USER ID = @UserIdNumber+Username+Password+EmailAddress+CreationDate

Suspicious USER = Title+Forename+Surname+DOB+Gender+ADDRESS

ADDRESS = HouseNumber+AddressLine1+ (AddressLine2) +City+County+PostCode+ (ZIPCode) +Country

CONTACT DETAILS = HomeTelephoneNumber+ (MobileTelephoneNumber) +EmailAddress

PROFILE = AboutMe+ Industry+ Software + Reason for suspicion

Data Elements

Element	Data	Format/Description
Suspicious USERID		
UserIdNumber	7{Integer}7	####### each # is [0…9**]**
Username	1{Char}20	
Password	1{Char}20	
EmailAddress	1{Char}20	
CreationDate	date	##-##-#### each # is [0…9]
Suspicious USER		
Title	1{char}4	Usually from [Mr|Mrs|Master|Miss]
Forename	1{char}20	
Surname	1{char}20	
DOB	date	##-##-#### each # is [0…9]
Gender	Boolean	List 0 = Male 1 = Female
ADDRESS		
HouseNumber	1{char}3	Usually numeric but can be subset [0…9|A…Z|a…z]
AddressLine1	1{char}30	
AddressLine2	1{char}30	
City	1{char}30	
County	1{char}30	
Postcode	7{char}7	Format $$# %$$ where$= [A…Z]

		#=[1..99] and %=[0..9]
ZIPCode	7{char}7	Format $$# %$$ where$= [A...Z] #=[1..99] and %=[0..9]
Country	1{char} 25	
CONTACT DETAILS		
HomeTelephoneNumber	integer	Varies due to phone length in different countries
MobileTelephoneNumber	integer	
EmailAddress	1{char}20	
PROFILE		
AboutMe	1{char}MAX	
Industry	[A...Z|a...z]	
Software	1{char}20	
Reason for suspicion	1{char}200	

3.1.1.11 Data Store Deleted Users

Data Structures:

Deleted USER ID = @UserIdNumber+Username+Password+EmailAddress+CreationDate

Deleted USER = Title+Forename+Surname+DOB+Gender+ADDRESS

ADDRESS = HouseNumber+AddressLine1+ (AddressLine2) +City+County+PostCode+ (ZIPCode) +Country

CONTACT DETAILS = HomeTelephoneNumber+ (MobileTelephoneNumber) +EmailAddress

PROFILE = AboutMe+ Industry+ Software + Reason for suspicion

Data Elements

Element	Data	Format/Description
Deleted USERID		
UserIdNumber	7{Integer}7	####### each # is [0…9**]**
Username	1{Char}20	
Password	1{Char}20	
EmailAddress	1{Char}20	
CreationDate	date	##-##-#### each # is [0…9]
Deletion Data	Date	##-##-#### each # is [0…9]
Deleted USER		
Title	1{char}4	Usually from [Mr|Mrs|Master|Miss]
Forename	1{char}20	
Surname	1{char}20	
DOB	date	##-##-#### each # is [0…9]
Gender	Boolean	List 0 = Male 1 = Female
ADDRESS		
HouseNumber	1{char}3	Usually numeric but can be subset [0…9|A…Z|a…z]
AddressLine1	1{char}30	
AddressLine2	1{char}30	
City	1{char}30	
County	1{char}30	
Postcode	7{char}7	Format $$# %$$ where$= [A…Z] #=[1..99] and %=[0..9]
ZIPCode	7{char}7	Format $$# %$$ where$= [A…Z] #=[1..99] and %=[0..9]
Country	1{char} 25	
CONTACT DETAILS		
HomeTelephoneNumber	integer	Varies due to phone length in different countries
MobileTelephoneNumber	integer	
EmailAddress	1{char}20	
PROFILE		
AboutMe	1{char}MAX	
Industry	[A…Z|a…z]	
Software	1{char}20	
Reason for deletion	1{char}200	

3.1.1.12 Data Store Demo Application

Data structures:

Functionalities = FirstFunctionality + SecondFunctionality + NFunctionality

***Q.A* = Title**+Forename+Surname+DOB+Gender +ADDRESS

ADDRESS = HouseNumber+AddressLine1+ (AddressLine2)
+City+County+PostCode+ (ZIPCode) +Country

Q.A results = Results

***Software developer*= Title**+Forename+Surname+DOB+Gender +ADDRESS

ADDRESS = HouseNumber+AddressLine1+ (AddressLine2)
+City+County+PostCode+ (ZIPCode) +Country

***System Analysis* = Title**+Forename+Surname+DOB+Gender +ADDRESS

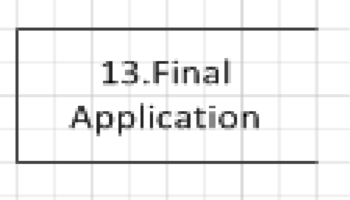
ADDRESS = HouseNumber+AddressLine1+ (AddressLine2)
+City+County+PostCode+ (ZIPCode) +Country

Data Elements

Element	Data	Format/Description
Functionalities		
FirstFunctionality	1{char}300	
SecondFunctionality	1{char}300	
NFunctionality	1{char}300	
Q.A		
Title	1{char}4	Usually from [Mr|Mrs|Master|Miss]
Forename		1{char}20
Surname		1{char}20
DOB	date	##-##-#### each # is [0…9]

Gender	Boolean	List 0 = Male 1 = Female
ADDRESS		
HouseNumber	1{char}3	Usually numeric but can be subset [0...9|A...Z|a...z]
AddressLine1	1{char}30	
AddressLine2	1{char}30	
City	1{char}30	
County	1{char}30	
Postcode	7{char}7	Format $$# %$$ where$= [A...Z] #=[1..99] and %=[0..9]
Quality Assurance result		
Results	Boolean	0 – Failed, 1 Passed
Software developer		
Title	1{char}4	Usually from [Mr|Mrs|Master|Miss]
Forename	1{char}20	
Surname	1{char}20	
DOB	date	##-##-#### each # is [0...9]
Gender	Boolean	List 0 = Male 1 = Female
ADDRESS		
HouseNumber	1{char}3	Usually numeric but can be subset [0...9|A...Z|a...z]
AddressLine1	1{char}30	
AddressLine2	1{char}30	
City	1{char}30	
County	1{char}30	
Postcode	7{char}7	Format $$# %$$ where$= [A...Z] #=[1..99] and %=[0..9]
System Analysis		
Title	1{char}4	Usually from [Mr|Mrs|Master|Miss]
Forename	1{char}20	
Surname	1{char}20	
DOB	date	##-##-#### each # is [0...9]
Gender	Boolean	List 0 = Male 1 = Female
ADDRESS		
HouseNumber	1{char}3	Usually numeric but can be subset [0...9|A...Z|a...z]
AddressLine1	1{char}30	
AddressLine2	1{char}30	
City	1{char}30	
County	1{char}30	
Postcode	7{char}7	Format $$# %$$ where$= [A...Z] #=[1..99] and %=[0..9]

3.1.1.13 Data Store Final Application



Data structures:

Functionalities = FirstFunctionality + SecondFunctionality + NFunctionality

***Q.A* = Title**+Forename+Surname+DOB+Gender +ADDRESS

ADDRESS = HouseNumber+AddressLine1+ (AddressLine2)
+City+County+PostCode+ (ZIPCode) +Country

Q.A results = Results

***Software developer*= Title**+Forename+Surname+DOB+Gender +ADDRESS

ADDRESS = HouseNumber+AddressLine1+ (AddressLine2)
+City+County+PostCode+ (ZIPCode) +Country

***System Analysis* = Title**+Forename+Surname+DOB+Gender +ADDRESS

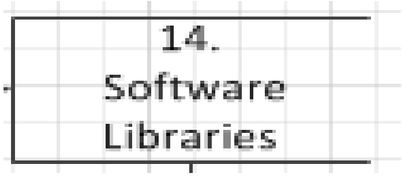
ADDRESS = HouseNumber+AddressLine1+ (AddressLine2)
+City+County+PostCode+ (ZIPCode) +Country

Data Elements

Element	**Data**	**Format/Description**
Functionalities		
FirstFunctionality	1{char}300	
SecondFunctionality	1{char}300	
NFunctionality	1{char}300	
Q.A		
Title	1{char}4	Usually from [Mr|Mrs|Master|Miss]
Forename	1{char}20	
Surname	1{char}20	
DOB	date	##-##-#### each # is [0…9]

Gender	Boolean	List 0 = Male 1 = Female
ADDRESS		
HouseNumber	1{char}3	Usually numeric but can be subset [0...9|A...Z|a...z]
AddressLine1	1{char}30	
AddressLine2	1{char}30	
City	1{char}30	
County	1{char}30	
Postcode	7{char}7	Format $$# %$$ where$= [A...Z] #=[1..99] and %=[0..9]
Quality Assurance result		
Results	Boolean	0 – Failed, 1 Passed
Software developer		
Title	1{char}4	Usually from [Mr|Mrs|Master|Miss]
Forename	1{char}20	
Surname	1{char}20	
DOB	date	##-##-#### each # is [0...9]
Gender	Boolean	List 0 = Male 1 = Female
ADDRESS		
HouseNumber	1{char}3	Usually numeric but can be subset [0...9|A...Z|a...z]
AddressLine1	1{char}30	
AddressLine2	1{char}30	
City	1{char}30	
County	1{char}30	
Postcode	7{char}7	Format $$# %$$ where$= [A...Z] #=[1..99] and %=[0..9]
System Analysis		
Title	1{char}4	Usually from [Mr|Mrs|Master|Miss]
Forename	1{char}20	
Surname	1{char}20	
DOB	date	##-##-#### each # is [0...9]
Gender	Boolean	List 0 = Male 1 = Female
ADDRESS		
HouseNumber	1{char}3	Usually numeric but can be subset [0...9|A...Z|a...z]
AddressLine1	1{char}30	
AddressLine2	1{char}30	
City	1{char}30	
County	1{char}30	
Postcode	7{char}7	Format $$# %$$ where$= [A...Z] #=[1..99] and %=[0..9]

3.1.1.14 Data Store Software Libraries



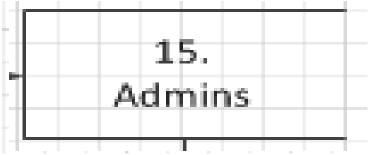
Data structures:

Type of Software Libraries = @Web development + Wed design + Mobile development + Microsoft Development + Java Development

Data Elements

Element	Data	Format/Description
Type of Software Libraries		
Web development	Boolean	0-false, 1-true
Wed design	Boolean	0-false, 1-true
Mobile development	Boolean	0-false, 1-true
Microsoft development	Boolean	0-false, 1-true
Java development	Boolean	0-false, 1-true

3.1.1.15 Data Store Admins



Data structures:

Area of Expertise = @Web Site + Software development System + System administration

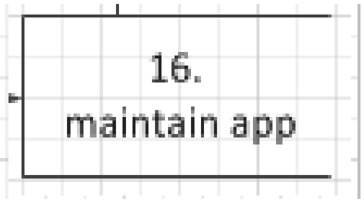
Admin = Title+Forename+Surname+DOB+Gender + Documented System requirements +ADDRESS

ADDRESS = HouseNumber+AddressLine1+ (AddressLine2) +City+County+PostCode+ (ZIPCode) +Country

Data Elements

Element	Data	Format/Description
Area of Expertise		
Web Site	Boolean	0-false, 1-true
Software development System	Boolean	0-false, 1-true
System administrations	Boolean	0-false, 1-true
Admin		
Title	1{char}4	Usually from [Mr|Mrs|Master|Miss]
Forename	1{char}20	
Surname	1{char}20	
DOB	date	##-##-#### each # is [0...9]
Gender	Boolean	List 0 = Male 1 = Female
ADDRESS		
HouseNumber	1{char}3	Usually numeric but can be subset [0...9|A...Z|a...z]
AddressLine1	1{char}30	
AddressLine2	1{char}30	
City	1{char}30	
County	1{char}30	
Postcode	7{char}7	Format $$# %$$ where$= [A...Z] #=[1..99] and %=[0..9]

3.1.1.16 Data Store Maintain app



Data structures:

Area of Maintaining = @Web development + Wed design + Mobile development + Microsoft Development + Java Developement

Software Developer = Title+Forename+Surname+DOB+Gender + Documented System requirements +ADDRESS

Manager = Title+Forename+Surname+DOB+Gender + Documented System requirements +ADDRESS

Data Elements

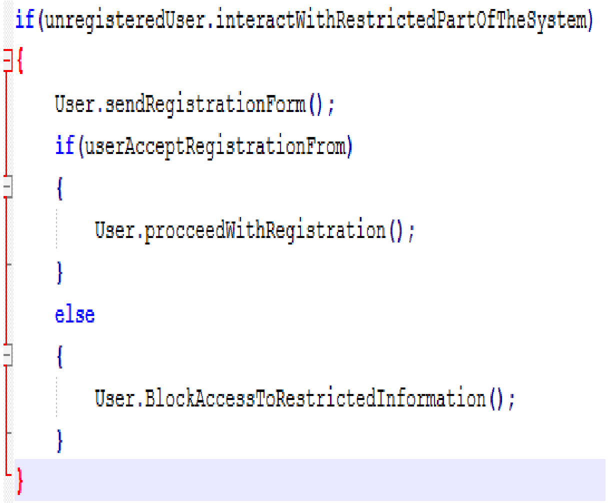
Element	Data	Format/Description
Area of Maintaining		
Web development	Boolean	0-false, 1-true
Wed design	Boolean	0-false, 1-true
Mobile development	Boolean	0-false, 1-true
Microsoft development	Boolean	0-false, 1-true
Java development	Boolean	0-false, 1-true
Software Developer		
Title	1{char}4	Usually from [Mr|Mrs|Master|Miss]
Forename	1{char}20	
Surname	1{char}20	
DOB	date	##-##-#### each # is [0…9]
Gender	Boolean	List 0 = Male 1 = Female
Manager		
Title	1{char}4	Usually from [Mr|Mrs|Master|Miss]
Forename	1{char}20	
Surname	1{char}20	
DOB	date	##-##-#### each # is [0…9]
Gender	Boolean	List 0 = Male 1 = Female

3.1.2 Process descriptors

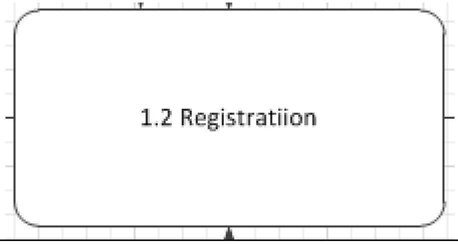
The following process descriptors show how something is to be done within a process. Each process will be described from the Top level data flow diagram, as the lower level diagram processes are too detailed. In the descriptor it will describe the process in a few lines and write what inputs and outputs it has to deal with from both terminators and data stores. It will then be gone on to write if the process has any variables and what the mini specification for the process will be. The mini spec is where it will be described the action in terms or algorithms or structured English. I will describe the process in a structure English paragraph and use the algorithmic method underneath. The paragraph above should strengthen it and give people that are not familiar with algorithms a better comprehension.

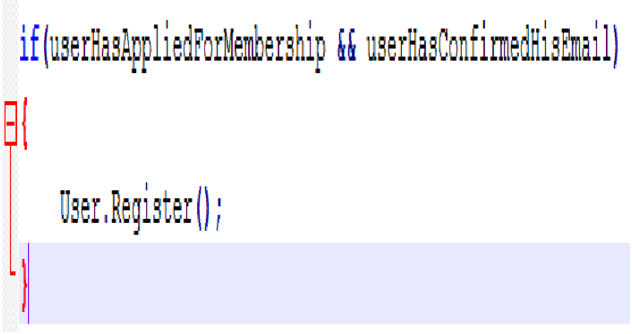
3.1.2.1 Process descriptors - Manage unregistered users

DATA DICTIONARY:PROCESS DESCRIPTORS	
Name:	1.1 Manage Unregistered Users
Description:	Managing users who aren't registered to the system and getting them to register via a registration request
Data Inputs	Dataflow interaction from terminator (a unregistered user) Dataflow Register from terminator (a unregistered user)
Data Outputs	Dataflow Registration request to terminator (a Unregistered user) Dataflow Registration confirmation to terminator (a Unregistered user) Dataflow New user created to data store 1.User
Variables	N/A

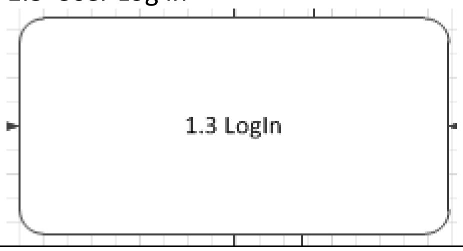
Mini-Spec	If the unregistered user interacts with a restricted part of the system, (THEN) send a registration request to the user. If the user responds to the registration request and registers, (THEN) create a new user from the unregistered user's registration details. This information is then added to the User data store
	
	`if(unregisteredUser.interactWithRestrictedPartOfTheSystem)`
	`{`
	`    User.sendRegistrationForm();`
	`    if(userAcceptRegistrationFrom)`
	`    {`
	`        User.procceedWithRegistration();`
	`    }`
	`    else`
	`    {`
	`        User.BlockAccessToRestrictedInformation();`
	`    }`
	`}`

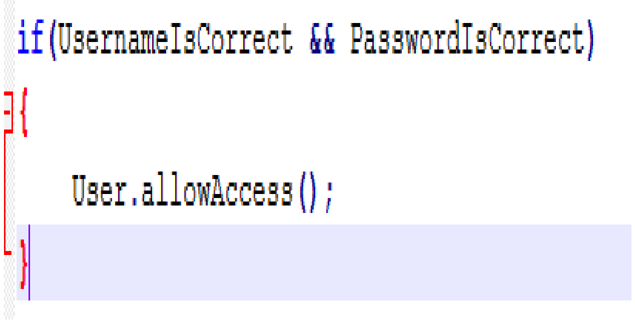
3.1.2.2 Process descriptors – Registration

DATA DICTIONARY:PROCESS DESCRIPTORS	
Name:	1.2 Registration 
Description:	Registration the users into the system.
Data Inputs	Dataflow Log in attempt from terminator (a User) Dataflow User details from data store 1.User
Data Outputs	Dataflow Attempt Accepted/Rejected to terminator (a User)
Variables	Password change

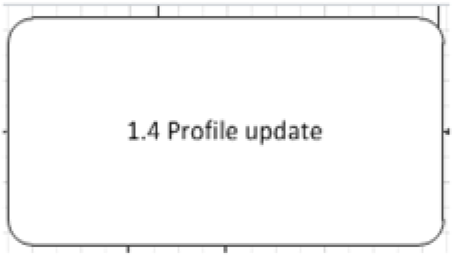
Mini-**Spec**	If the user has applied for membership he will be send an email for confirmation. When the user clicks on the link for the confirmation his user will be activated and his registration completed.
	
	`if(userHasAppliedForMembership && userHasConfirmedHisEmail)`
	`{`
	`    User.Register();`
	`}`

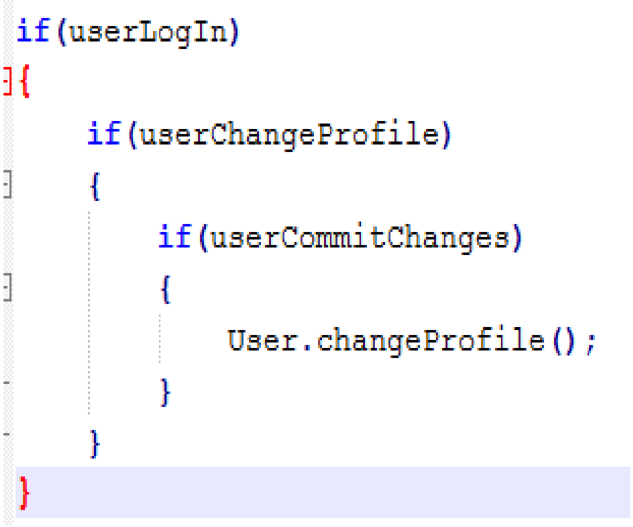
3.1.2.3 Process descriptors – User log in

DATA DICTIONARY:PROCESS DESCRIPTORS	
Name:	1.3 User Log in 
Description:	Processing the users log in details and recording them to a Log Cat data store ready for the admin to review
Data Inputs	Dataflow Log in attempt from terminator (a User) Dataflow User details from data store 1.User
Data Outputs	Dataflow Attempt Accepted/Rejected to terminator (a User) Dataflow Log in recorded to data store 2.Log Cat
Variables	Password change

Mini-**Spec**	If the user attempts to log in using a username and password, read from the user data store to cross reference the username and password to see if it's correct. If they are both correct allow the user in the system. If they are incorrect, deny access to the user and ask them to try again. The results are then added to the Log in audit data store.
	
	`if(UsernameIsCorrect && PasswordIsCorrect)`
	`{`
	`    User.allowAccess();`
	`}`

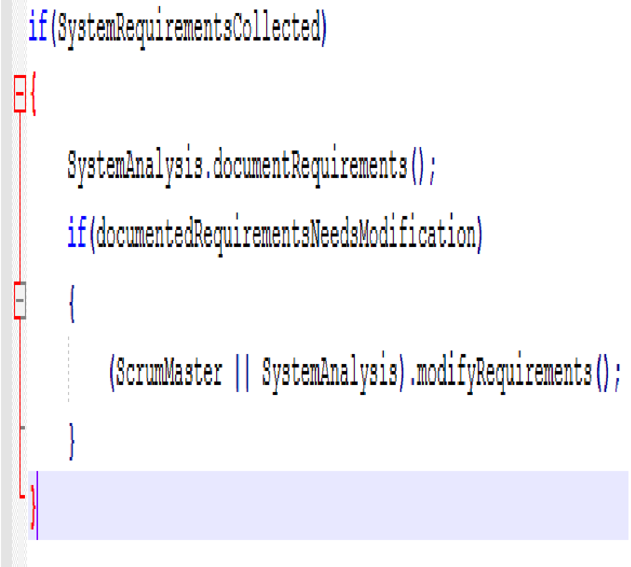
3.1.2.4 Process descriptors – Profile update

DATA DICTIONARY:PROCESS DESCRIPTORS	
Name:	1.4 Profile Update 
Description:	Managing registered users updates
Data Inputs	Dataflow Log in attempt from terminator (b User) Dataflow User details from data store 1.User
Data Outputs	Dataflow Attempt Accepted/Rejected to terminator (a User) Dataflow Log in recorded to data store 2.LogCat
Variables	Password change

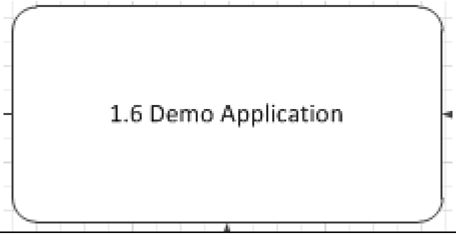
Mini-**Spec**	If the user is log in then he will have the option to change his profile data. If the selects to change his profile data and commit it then the profile data will be changed.
	
	`if(userLogIn)`
	`{`
	`    if(userChangeProfile)`
	`    {`
	`        if(userCommitChanges)`
	`        {`
	`            User.changeProfile();`
	`        }`
	`    }`
	`}`

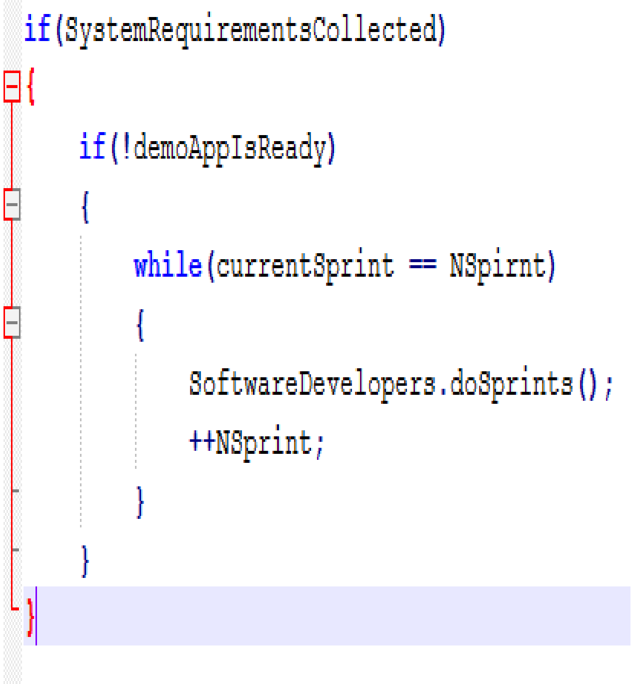
3.1.2.5 Process descriptors –System Requirements

DATA DICTIONARY:PROCESS DESCRIPTORS	
Name:	1.5System Requirements
Description:	System Analysis collects the system requirements from the user and then documents them to become official documents.
Data Inputs	Dataflow input from 7.System Analysis Dataflow input from 9.Scrum Master
Data Outputs	Dataflow output to the 3.System Requirements
Variables	Document system requirements

Mini-**Spec**	The System Analysis meets with the user and documents the requirements. If the user does not like the documented requirements the system analysis modifies it with the help of a scrum master.
	
	`if(SystemRequirementsCollected)`
	`{`
	`    SystemAnalysis.documentRequirements();`
	`    if(documentedRequirementsNeedsModification)`
	`    {`
	`        (ScrumMaster || SystemAnalysis).modifyRequirements();`
	`    }`
	`}`

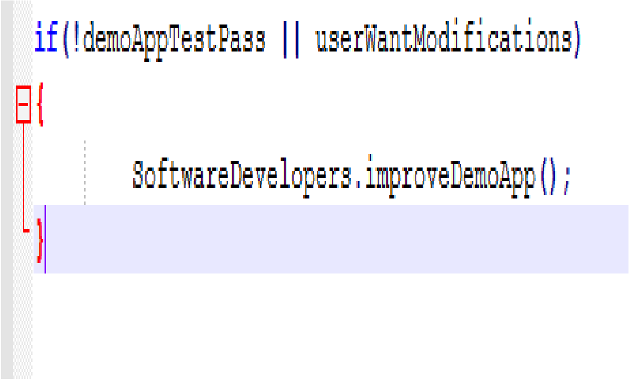
3.1.2.6Process descriptors –Demo Application

DATA DICTIONARY:PROCESS DESCRIPTORS	
Name:	1.6 Demo Application 
Description:	The Demo Application is developed by the software developers with a few sprints. This is done in order to be shown to the user and from then on the final application can be modified and meet the user expectations
Data Inputs	Dataflow input from 6. Software developers
Data Outputs	Dataflow output to the 3.System Requirements
Variables	N/A

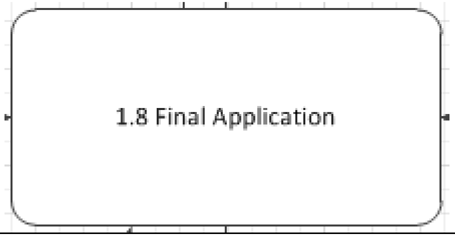
Mini-**Spec**	The Software developers are doing a few sprints with the system requirements until the demo application is ready.
	
	if(SystemRequirementsCollected)
	{
	    if(!demoAppIsReady)
	    {
	        while(currentSprint == NSpirnt)
	        {
	            SoftwareDevelopers.doSprints();
	            ++NSprint;
	        }
	    }
	}

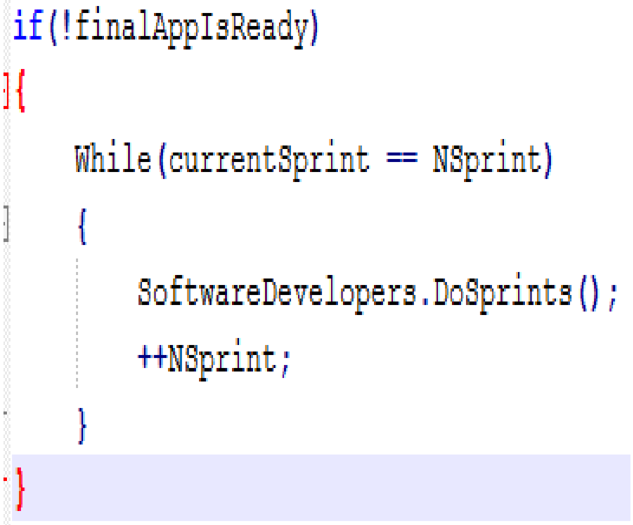
3.1.2.7 Process descriptors –Improve Demo Application

DATA DICTIONARY:PROCESS DESCRIPTORS	
Name:	1.7 Improve Demo Application
Description:	Improve Demo Application is improved by the software developers with a few corrections. This is done in order to satisfy the user or pass the tests of the Q.A.
Data Inputs	Dataflow input from 6. Software developers
Data Outputs	Dataflow output to the 3.System Requirements
Variables	N/A

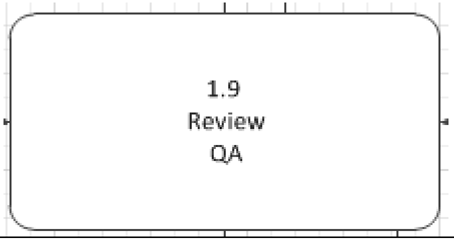
Mini-**Spec**	The Software developers are doing a few improvements until the tests pass and the user is happy with the demo application.
	
	`if(!demoAppTestPass || userWantModifications)`
	`{`
	`    SoftwareDevelopers.improveDemoApp();`
	`}`

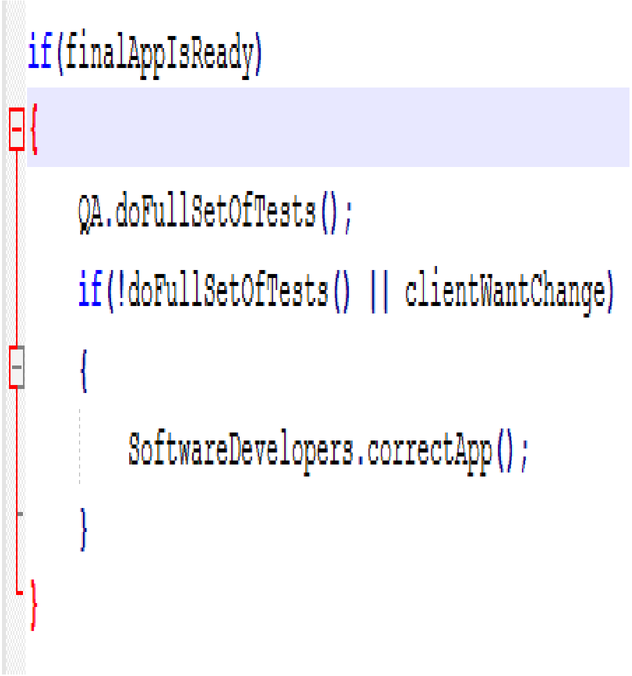
3.1.2.8Process descriptors –Final Application

DATA DICTIONARY:PROCESS DESCRIPTORS	
Name:	1.8 Final Application 
Description:	The Final Application is developed by the software developers with a couple of sprints. This is done in order to keep track of the progress of the application using agile software development methodology.
Data Inputs	Dataflow input from 6. Software developers
Data Outputs	Dataflow output to the 3.System Requirements
Variables	N/A

Mini-**Spec**	The Software developers are doing a couple of sprints with the system requirements until the final application is ready.
	
	if(!finalAppIsReady)
	{
	    While(currentSprint == NSprint)
	    {
	        SoftwareDevelopers.DoSprints();
	        ++NSprint;
	    }
	}

3.1.2.9 Process descriptors –Test the Application

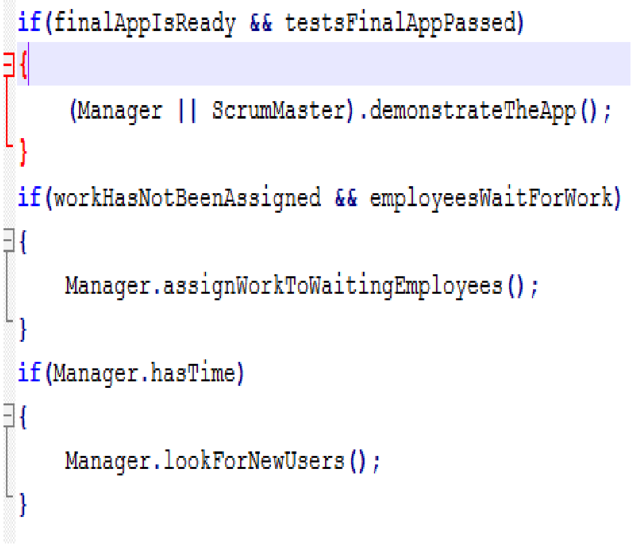
DATA DICTIONARY:PROCESS DESCRIPTORS	
Name:	1.9 Testing the app 
Description:	When the final application is done then a complete set of tests are done by the Q.A in order to make sure that everything is working as expected.
Data Inputs	Dataflow input from 4.QA Reports Dataflow input from 5. Q.A Dataflow input from 7. System Analysis
Data Outputs	Dataflow output from 4.QA Reports Dataflow output from 5. Q.A Dataflow output from 7. System Analysis
Variables	N/A

Mini-**Spec**	Q.As are doing a full set of testing's to guarantee the correctness of the application. If any test fails then this part is sent to the software developers to correct it.
	
	`if(finalAppIsReady)`
	`{`
	`    QA.doFullSetOfTests();`
	`    if(!doFullSetOfTests() || clientWantChange)`
	`    {`
	`        SoftwareDevelopers.correctApp();`
	`    }`
	`}`

3.1.2.10 Process descriptors –Manage the application

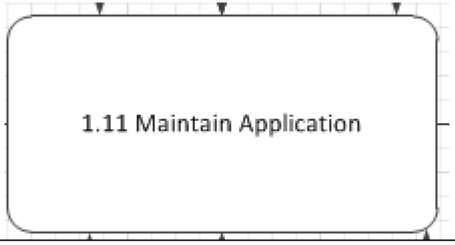
DATA DICTIONARY: PROCESS DESCRIPTORS

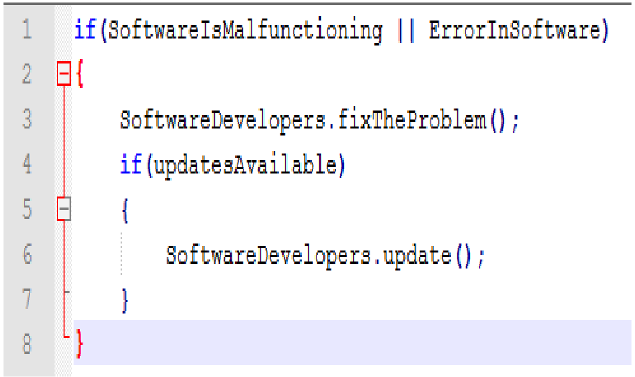
Name:	1.10 Manage the application
Description:	When the final application is done and approved then the manager closes the deal with the scrum master.
Data Inputs	Dataflow input from 8.Scrum Master Dataflow input from 6. Software developers Dataflow input from 9. Manager
Data Outputs	Dataflow output from 8.Scrum Master Dataflow output from 6. Software developers Dataflow output from 9. Manager
Variables	N/A

Mini-**Spec**	When the development and the test of the application are finished the scrum master or the manager present the masterpiece to the user and finalize the deal. The manager is also responsible to assign work to the employees and make sure that they are working on high standards and on time. The manager is looking for new users if there is time for that.
	
	`if(finalAppIsReady && testsFinalAppPassed)`
	`{`
	`    (Manager || ScrumMaster).demonstrateTheApp();`
	`}`
	`if(workHasNotBeenAssigned && employeesWaitForWork)`
	`{`
	`    Manager.assignWorkToWaitingEmployees();`
	`}`
	`if(Manager.hasTime)`
	`{`
	`    Manager.lookForNewUsers();`
	`}`

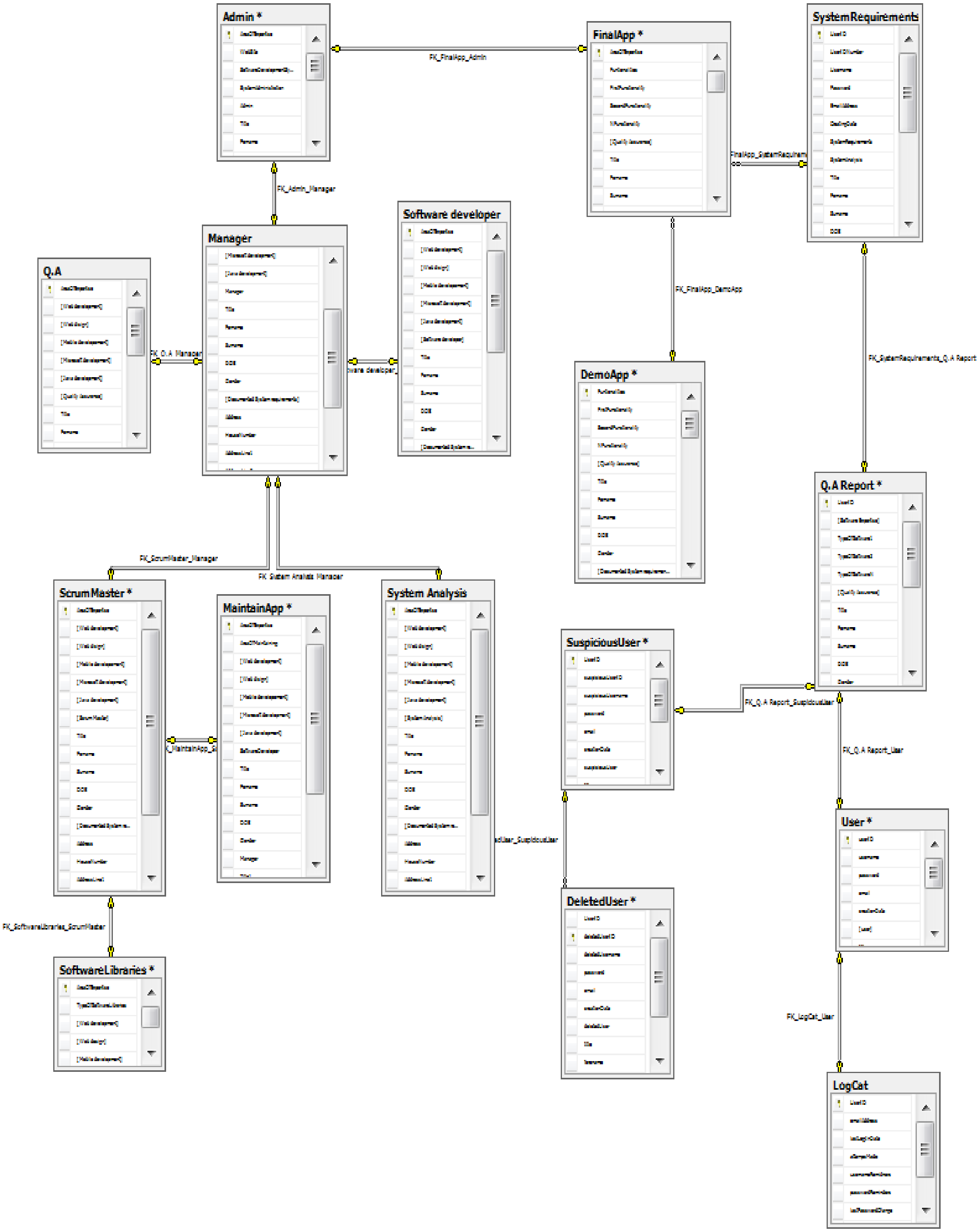
3.1.2.11 Process descriptors –Maintain the application

DATA DICTIONARY: PROCESS DESCRIPTORS

Name:	1.11 Maintain Application 
Description:	When the final application is done and approved then the manager closes the deal with the scrum master.
Data Inputs	Dataflow input from 1.User Dataflow input from 6. Software developers Dataflow input from 8.Scrum Master Dataflow input from 13.final app Dataflow input from 14.Sotware Libraries Dataflow input from 15.Admins Dataflow input from 16.Maintain app
Data Outputs	Dataflow output from 8.Scrum Master Dataflow output from 9. Manager Dataflow output from 15.Admins Dataflow output from 16.Maintain app
Variables	N/A

Mini-**Spec**	The Users can use the services of the system to maintain their software. This software can be written by employees to the system or others. The Software Developers and the Managers are mainly responsible to keep the software up to date and running without disturbances.
	
	`if(SoftwareIsMalfunctioning || ErrorInSoftware)`
	`{`
	`    SoftwareDevelopers.fixTheProblem();`
	`    if(updatesAvailable)`
	`    {`
	`        SoftwareDevelopers.update();`
	`    }`
	`}`

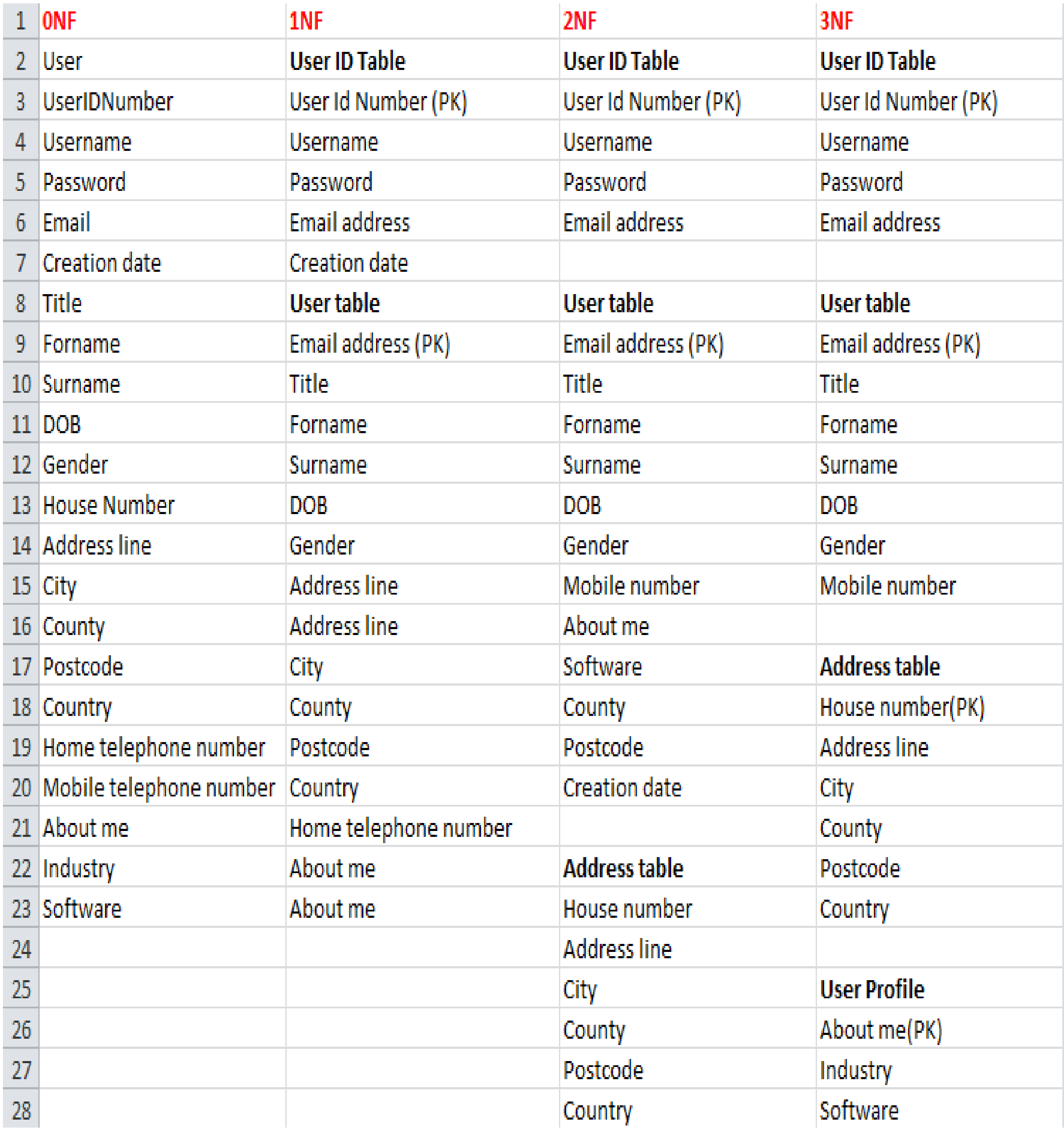
3.2 Data Model



3.3 Normalisation

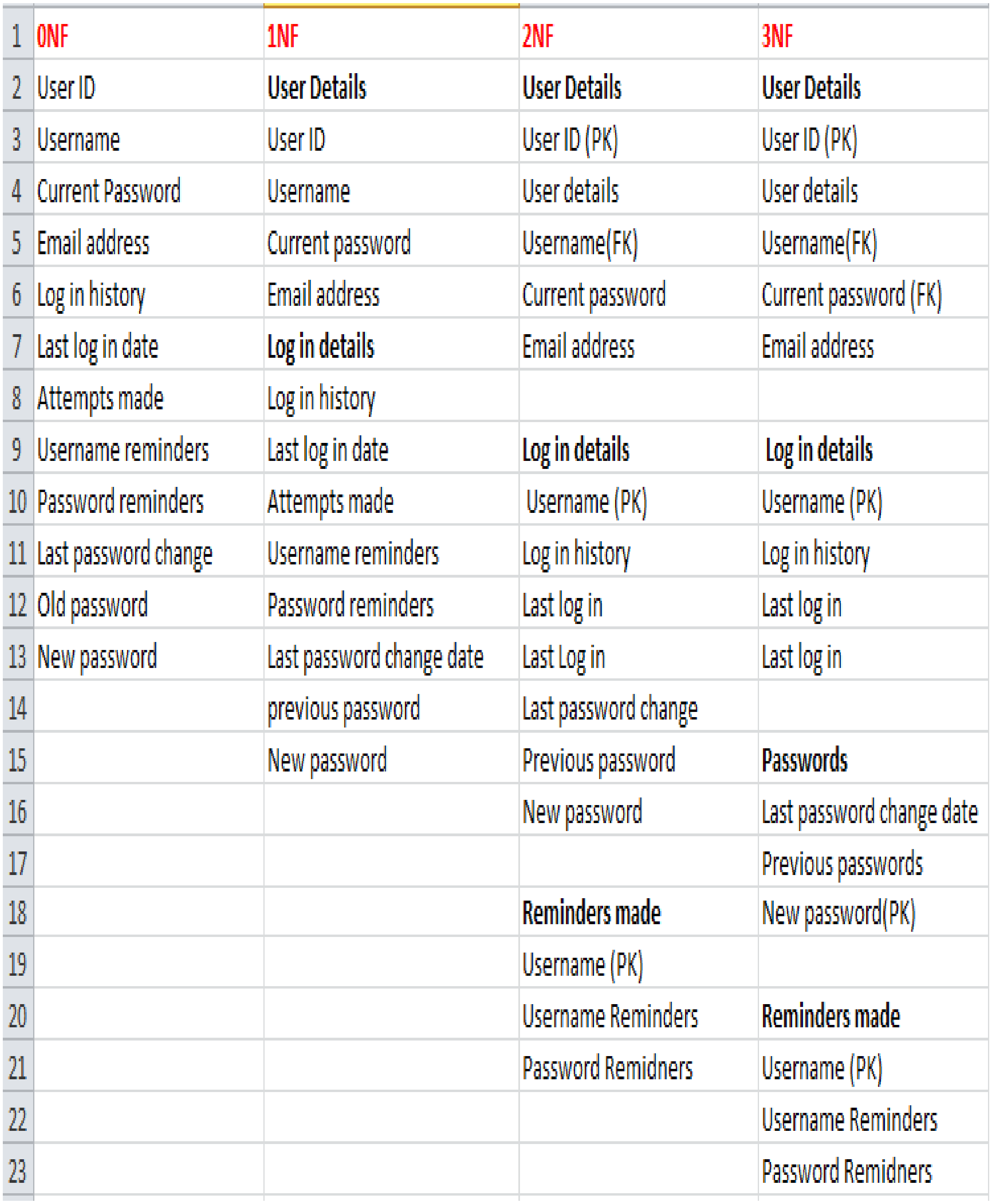
The normalisation stage is a step by step technique which restructures the data from the system in to a more efficient form. It takes the logical data store view to a more physical table/ database view. It removes the duplication in the data stores, and removes all nulls. It reduces the size of the data by removing key and non-key dependencies. It will eventually lead to fully optimised set of tables for implementation.

3.3.1 User



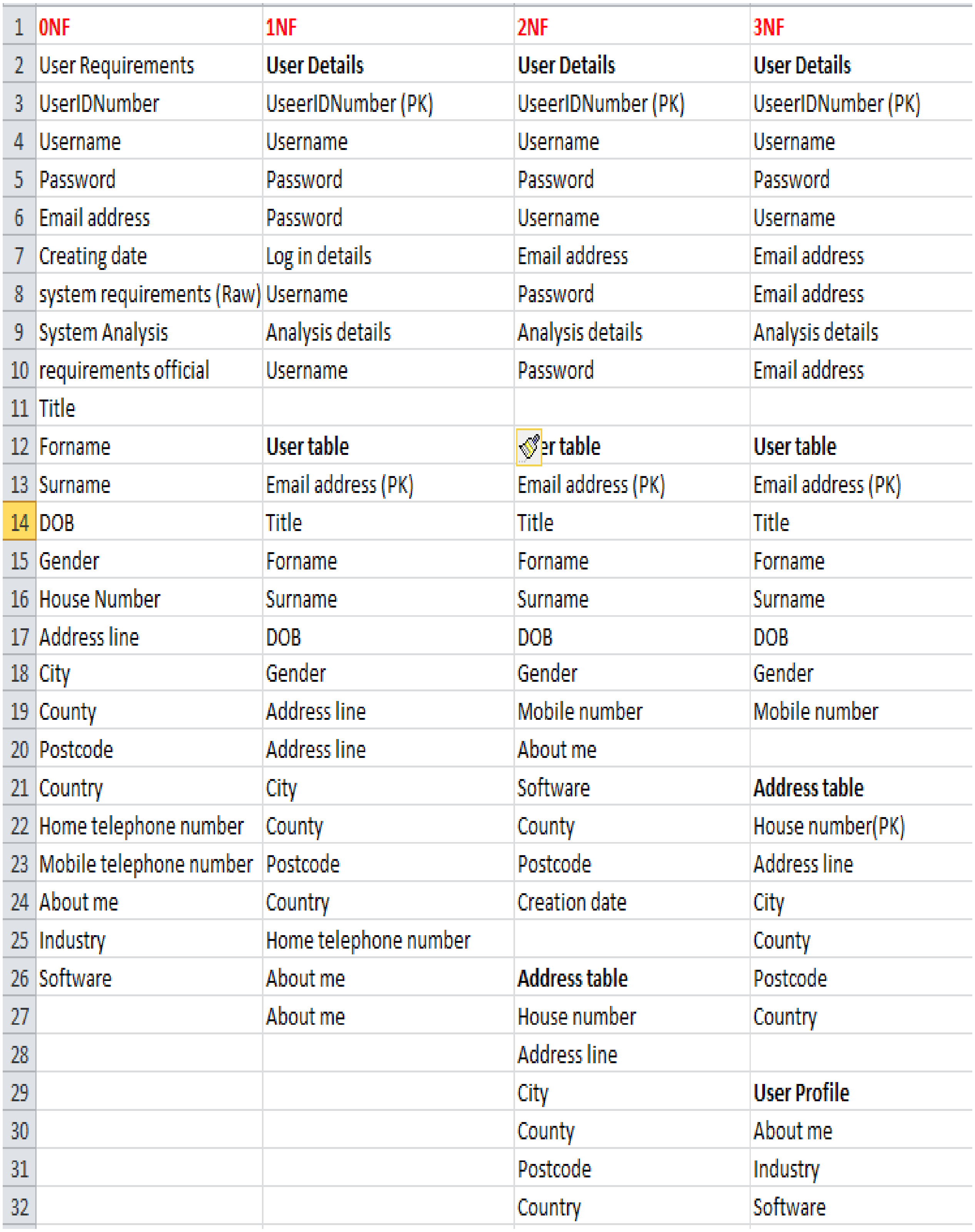
1	0NF	1NF	2NF	3NF
2	User	**User ID Table**	**User ID Table**	**User ID Table**
3	UserIDNumber	User Id Number (PK)	User Id Number (PK)	User Id Number (PK)
4	Username	Username	Username	Username
5	Password	Password	Password	Password
6	Email	Email address	Email address	Email address
7	Creation date	Creation date		
8	Title	**User table**	**User table**	**User table**
9	Forname	Email address (PK)	Email address (PK)	Email address (PK)
10	Surname	Title	Title	Title
11	DOB	Forname	Forname	Forname
12	Gender	Surname	Surname	Surname
13	House Number	DOB	DOB	DOB
14	Address line	Gender	Gender	Gender
15	City	Address line	Mobile number	Mobile number
16	County	Address line	About me	
17	Postcode	City	Software	**Address table**
18	Country	County	County	House number(PK)
19	Home telephone number	Postcode	Postcode	Address line
20	Mobile telephone number	Country	Creation date	City
21	About me	Home telephone number		County
22	Industry	About me	**Address table**	Postcode
23	Software	About me	House number	Country
24			Address line	
25			City	**User Profile**
26			County	About me(PK)
27			Postcode	Industry
28			Country	Software

3.3.2 LogCat



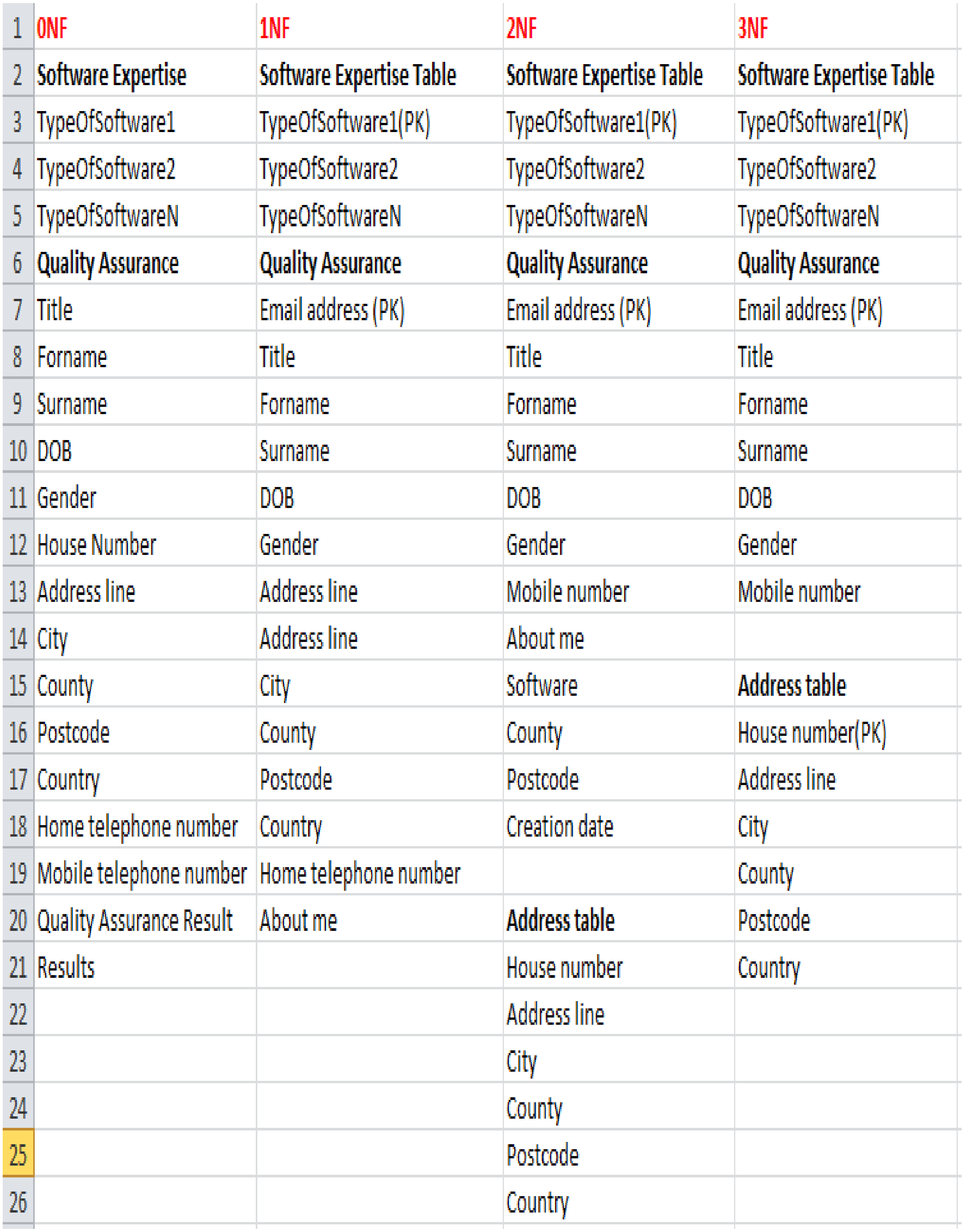
1	0NF	1NF	2NF	3NF
2	User ID	**User Details**	**User Details**	**User Details**
3	Username	User ID	User ID (PK)	User ID (PK)
4	Current Password	Username	User details	User details
5	Email address	Current password	Username(FK)	Username(FK)
6	Log in history	Email address	Current password	Current password (FK)
7	Last log in date	**Log in details**	Email address	Email address
8	Attempts made	Log in history		
9	Username reminders	Last log in date	**Log in details**	**Log in details**
10	Password reminders	Attempts made	Username (PK)	Username (PK)
11	Last password change	Username reminders	Log in history	Log in history
12	Old password	Password reminders	Last log in	Last log in
13	New password	Last password change date	Last Log in	Last log in
14		previous password	Last password change	
15		New password	Previous password	**Passwords**
16			New password	Last password change date
17				Previous passwords
18			**Reminders made**	New password(PK)
19			Username (PK)	
20			Username Reminders	**Reminders made**
21			Password Remidners	Username (PK)
22				Username Reminders
23				Password Remidners

3.3.3 System Requirements



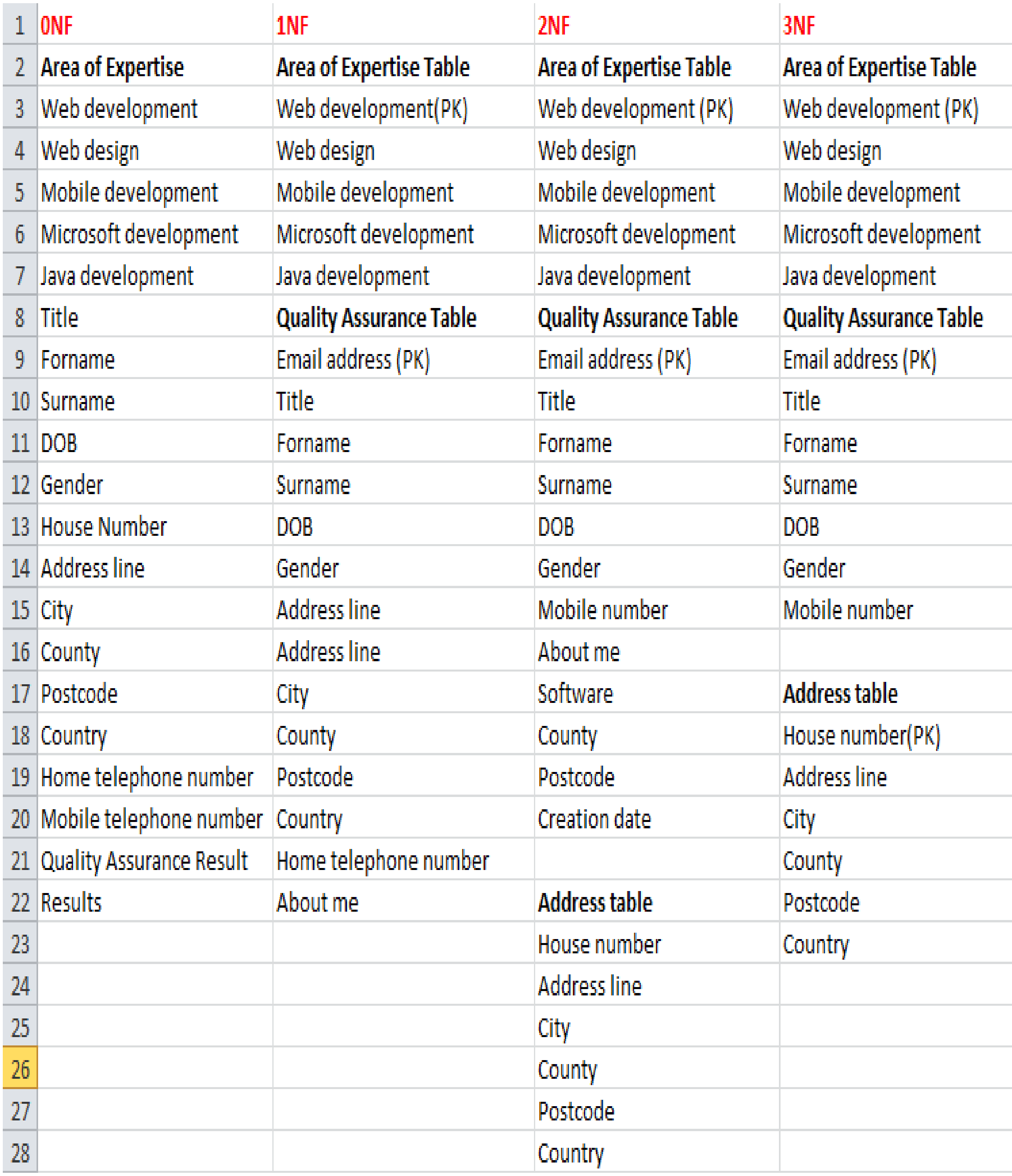
1	0NF	1NF	2NF	3NF
2	User Requirements	**User Details**	**User Details**	**User Details**
3	UserIDNumber	UseerIDNumber (PK)	UseerIDNumber (PK)	UseerIDNumber (PK)
4	Username	Username	Username	Username
5	Password	Password	Password	Password
6	Email address	Password	Username	Username
7	Creating date	Log in details	Email address	Email address
8	system requirements (Raw)	Username	Password	Email address
9	System Analysis	Analysis details	Analysis details	Analysis details
10	requirements official	Username	Password	Email address
11	Title			
12	Forname	**User table**	**er table**	**User table**
13	Surname	Email address (PK)	Email address (PK)	Email address (PK)
14	DOB	Title	Title	Title
15	Gender	Forname	Forname	Forname
16	House Number	Surname	Surname	Surname
17	Address line	DOB	DOB	DOB
18	City	Gender	Gender	Gender
19	County	Address line	Mobile number	Mobile number
20	Postcode	Address line	About me	
21	Country	City	Software	**Address table**
22	Home telephone number	County	County	House number(PK)
23	Mobile telephone number	Postcode	Postcode	Address line
24	About me	Country	Creation date	City
25	Industry	Home telephone number		County
26	Software	About me	**Address table**	Postcode
27		About me	House number	Country
28			Address line	
29			City	**User Profile**
30			County	About me
31			Postcode	Industry
32			Country	Software

3.3.4 Q.A report



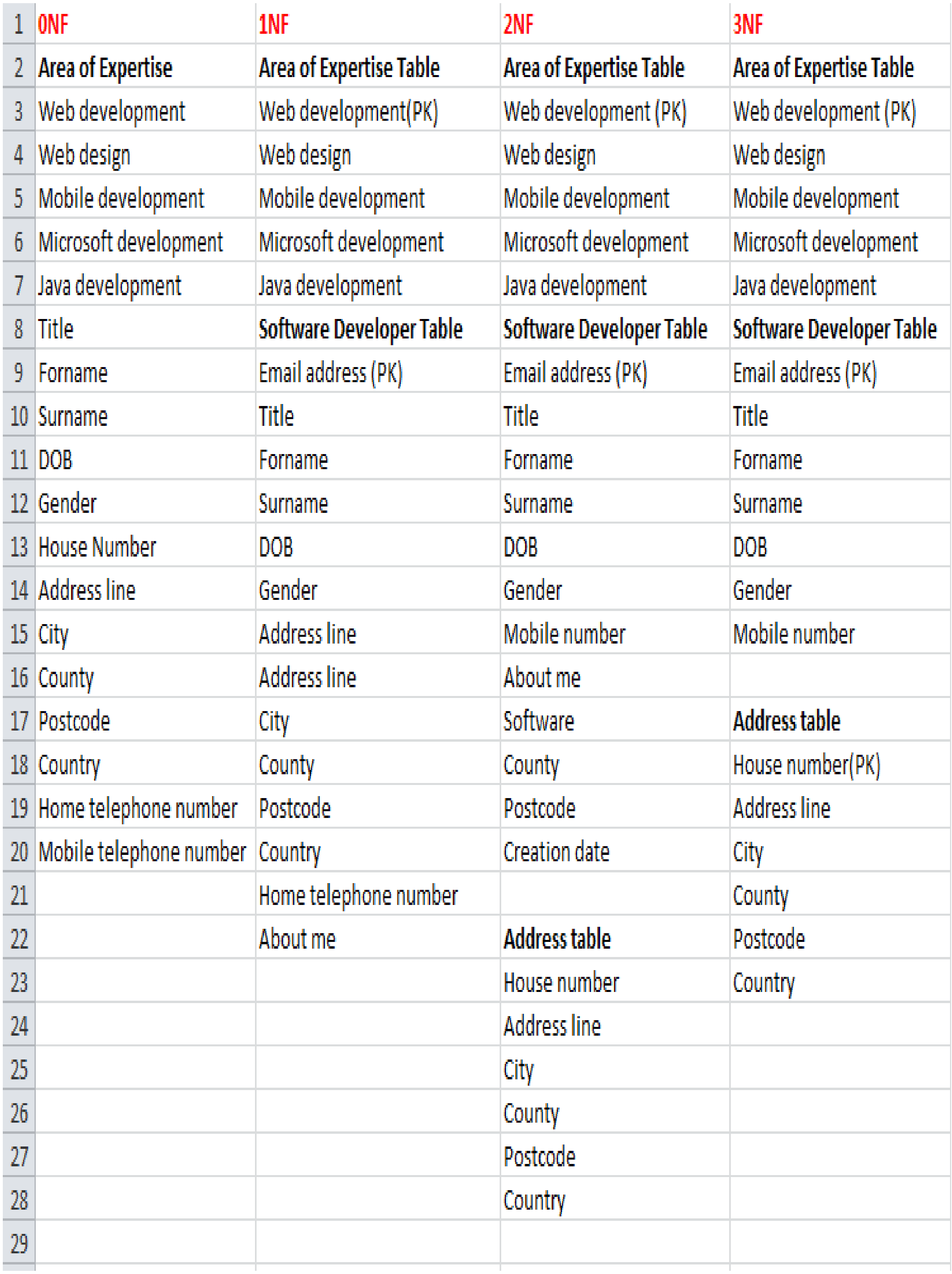
1	0NF	1NF	2NF	3NF
2	**Software Expertise**	**Software Expertise Table**	**Software Expertise Table**	**Software Expertise Table**
3	TypeOfSoftware1	TypeOfSoftware1(PK)	TypeOfSoftware1(PK)	TypeOfSoftware1(PK)
4	TypeOfSoftware2	TypeOfSoftware2	TypeOfSoftware2	TypeOfSoftware2
5	TypeOfSoftwareN	TypeOfSoftwareN	TypeOfSoftwareN	TypeOfSoftwareN
6	**Quality Assurance**	**Quality Assurance**	**Quality Assurance**	**Quality Assurance**
7	Title	Email address (PK)	Email address (PK)	Email address (PK)
8	Forname	Title	Title	Title
9	Surname	Forname	Forname	Forname
10	DOB	Surname	Surname	Surname
11	Gender	DOB	DOB	DOB
12	House Number	Gender	Gender	Gender
13	Address line	Address line	Mobile number	Mobile number
14	City	Address line	About me	
15	County	City	Software	**Address table**
16	Postcode	County	County	House number(PK)
17	Country	Postcode	Postcode	Address line
18	Home telephone number	Country	Creation date	City
19	Mobile telephone number	Home telephone number		County
20	Quality Assurance Result	About me	**Address table**	Postcode
21	Results		House number	Country
22			Address line	
23			City	
24			County	
25			Postcode	
26			Country	

3.3.5 Q.A



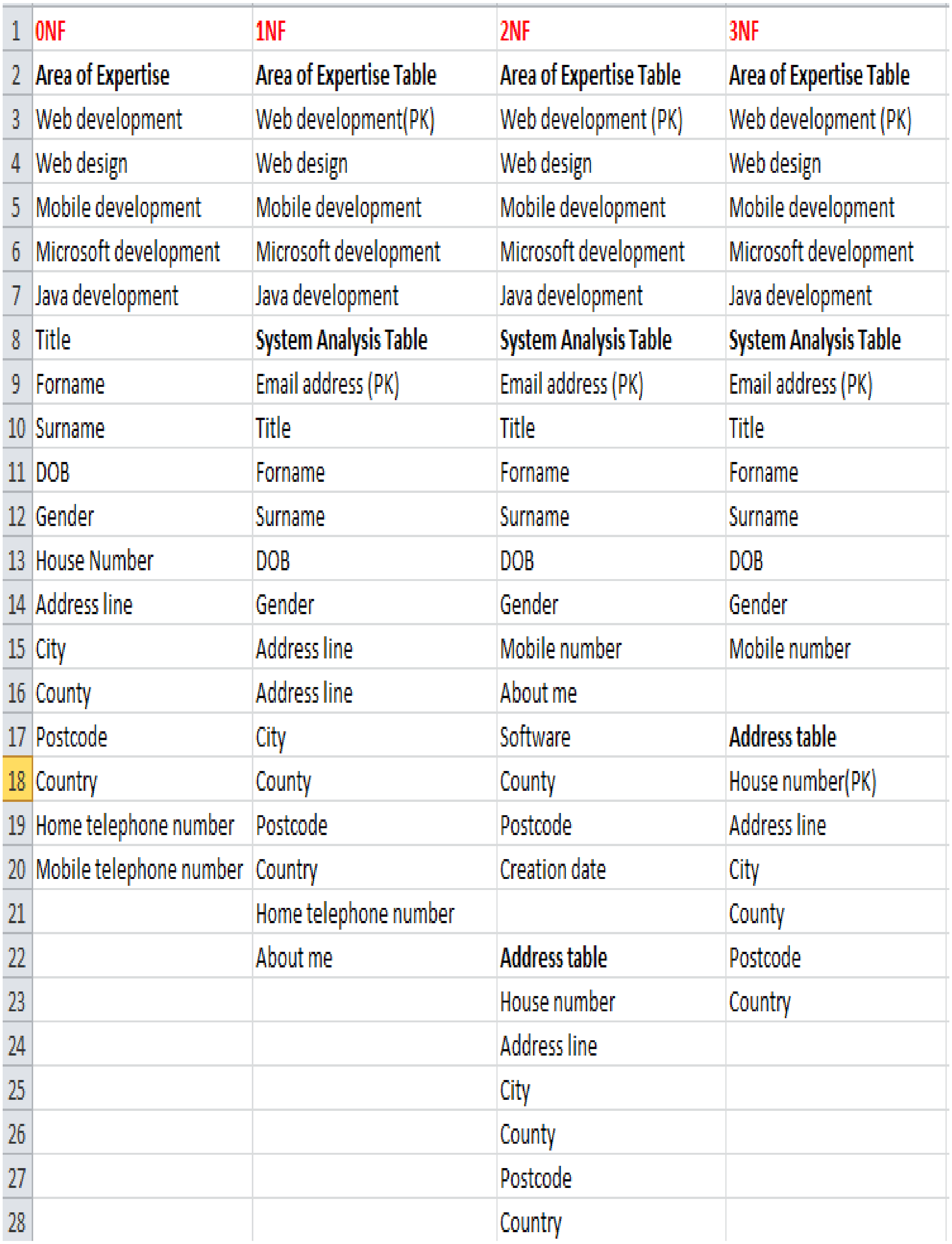
1	0NF	1NF	2NF	3NF
2	**Area of Expertise**	**Area of Expertise Table**	**Area of Expertise Table**	**Area of Expertise Table**
3	Web development	Web development(PK)	Web development (PK)	Web development (PK)
4	Web design	Web design	Web design	Web design
5	Mobile development	Mobile development	Mobile development	Mobile development
6	Microsoft development	Microsoft development	Microsoft development	Microsoft development
7	Java development	Java development	Java development	Java development
8	Title	**Quality Assurance Table**	**Quality Assurance Table**	**Quality Assurance Table**
9	Forname	Email address (PK)	Email address (PK)	Email address (PK)
10	Surname	Title	Title	Title
11	DOB	Forname	Forname	Forname
12	Gender	Surname	Surname	Surname
13	House Number	DOB	DOB	DOB
14	Address line	Gender	Gender	Gender
15	City	Address line	Mobile number	Mobile number
16	County	Address line	About me	
17	Postcode	City	Software	**Address table**
18	Country	County	County	House number(PK)
19	Home telephone number	Postcode	Postcode	Address line
20	Mobile telephone number	Country	Creation date	City
21	Quality Assurance Result	Home telephone number		County
22	Results	About me	**Address table**	Postcode
23			House number	Country
24			Address line	
25			City	
26			County	
27			Postcode	
28			Country	

3.3.6 Software Development



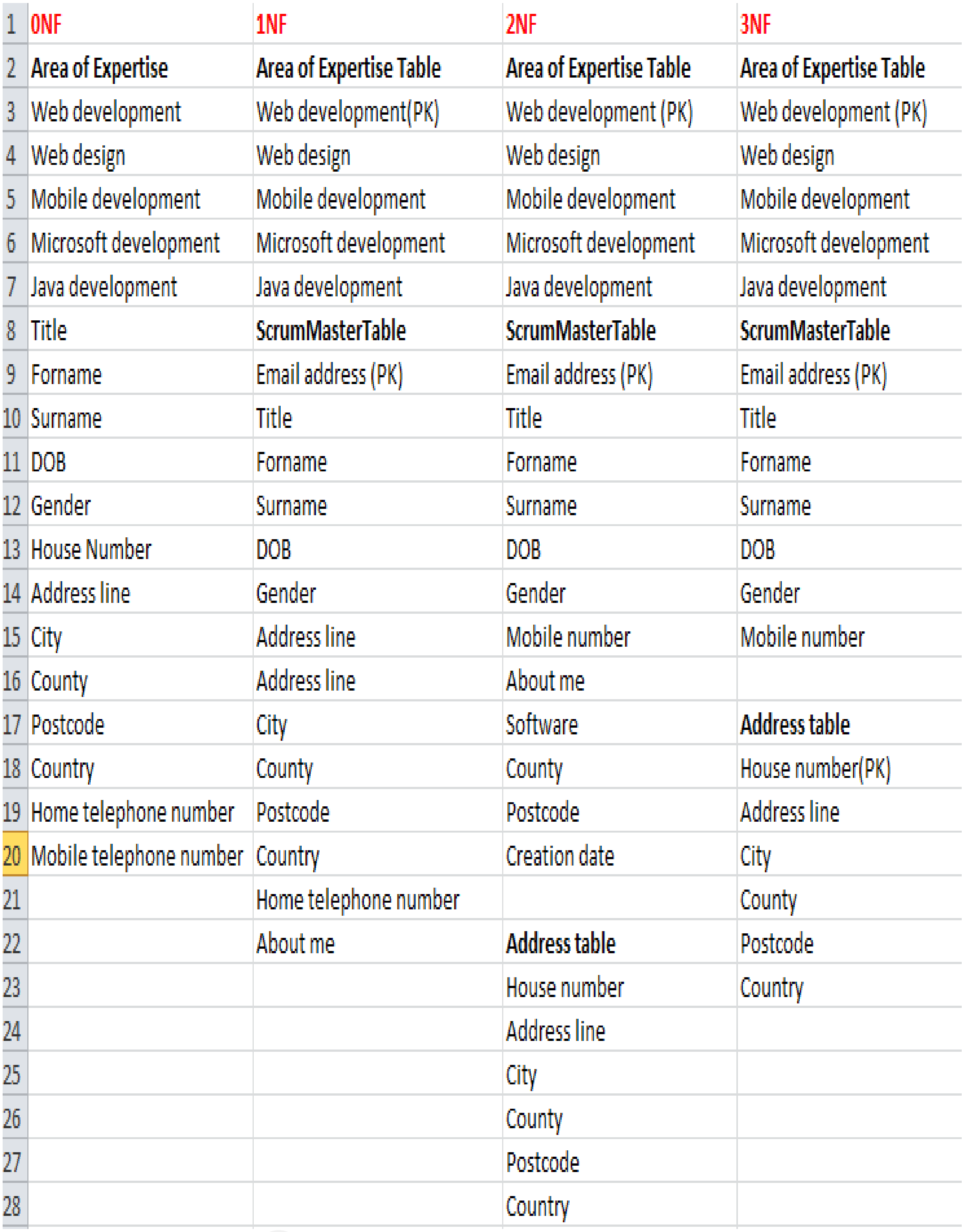
1	0NF	1NF	2NF	3NF
2	**Area of Expertise**	**Area of Expertise Table**	**Area of Expertise Table**	**Area of Expertise Table**
3	Web development	Web development(PK)	Web development (PK)	Web development (PK)
4	Web design	Web design	Web design	Web design
5	Mobile development	Mobile development	Mobile development	Mobile development
6	Microsoft development	Microsoft development	Microsoft development	Microsoft development
7	Java development	Java development	Java development	Java development
8	Title	**Software Developer Table**	**Software Developer Table**	**Software Developer Table**
9	Forname	Email address (PK)	Email address (PK)	Email address (PK)
10	Surname	Title	Title	Title
11	DOB	Forname	Forname	Forname
12	Gender	Surname	Surname	Surname
13	House Number	DOB	DOB	DOB
14	Address line	Gender	Gender	Gender
15	City	Address line	Mobile number	Mobile number
16	County	Address line	About me	
17	Postcode	City	Software	**Address table**
18	Country	County	County	House number(PK)
19	Home telephone number	Postcode	Postcode	Address line
20	Mobile telephone number	Country	Creation date	City
21		Home telephone number		County
22		About me	**Address table**	Postcode
23			House number	Country
24			Address line	
25			City	
26			County	
27			Postcode	
28			Country	
29				

3.3.7 System Analysis



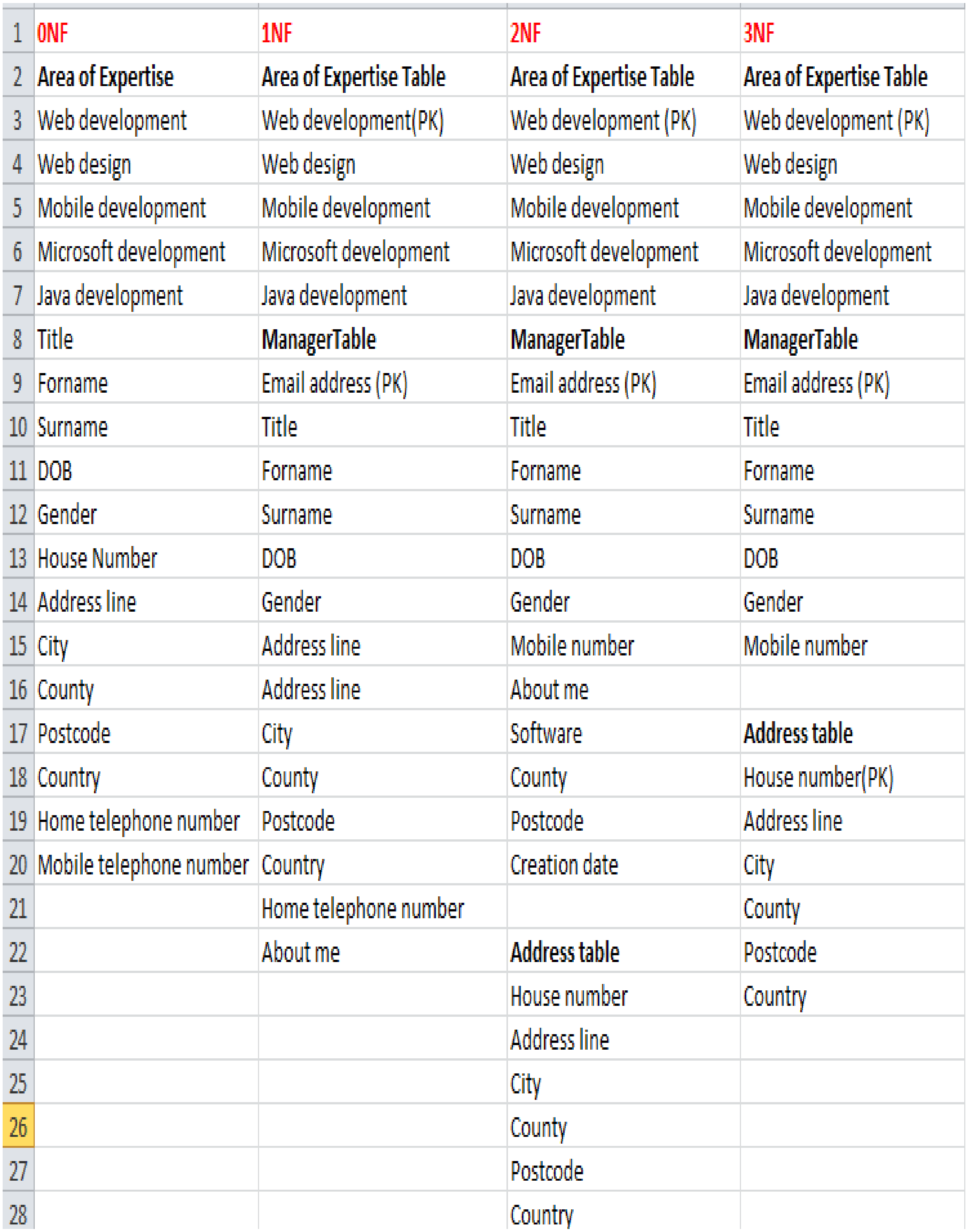
1	0NF	1NF	2NF	3NF
2	**Area of Expertise**	**Area of Expertise Table**	**Area of Expertise Table**	**Area of Expertise Table**
3	Web development	Web development(PK)	Web development (PK)	Web development (PK)
4	Web design	Web design	Web design	Web design
5	Mobile development	Mobile development	Mobile development	Mobile development
6	Microsoft development	Microsoft development	Microsoft development	Microsoft development
7	Java development	Java development	Java development	Java development
8	Title	**System Analysis Table**	**System Analysis Table**	**System Analysis Table**
9	Forname	Email address (PK)	Email address (PK)	Email address (PK)
10	Surname	Title	Title	Title
11	DOB	Forname	Forname	Forname
12	Gender	Surname	Surname	Surname
13	House Number	DOB	DOB	DOB
14	Address line	Gender	Gender	Gender
15	City	Address line	Mobile number	Mobile number
16	County	Address line	About me	
17	Postcode	City	Software	**Address table**
18	Country	County	County	House number(PK)
19	Home telephone number	Postcode	Postcode	Address line
20	Mobile telephone number	Country	Creation date	City
21		Home telephone number		County
22		About me	**Address table**	Postcode
23			House number	Country
24			Address line	
25			City	
26			County	
27			Postcode	
28			Country	

3.3.8 Scrum Master



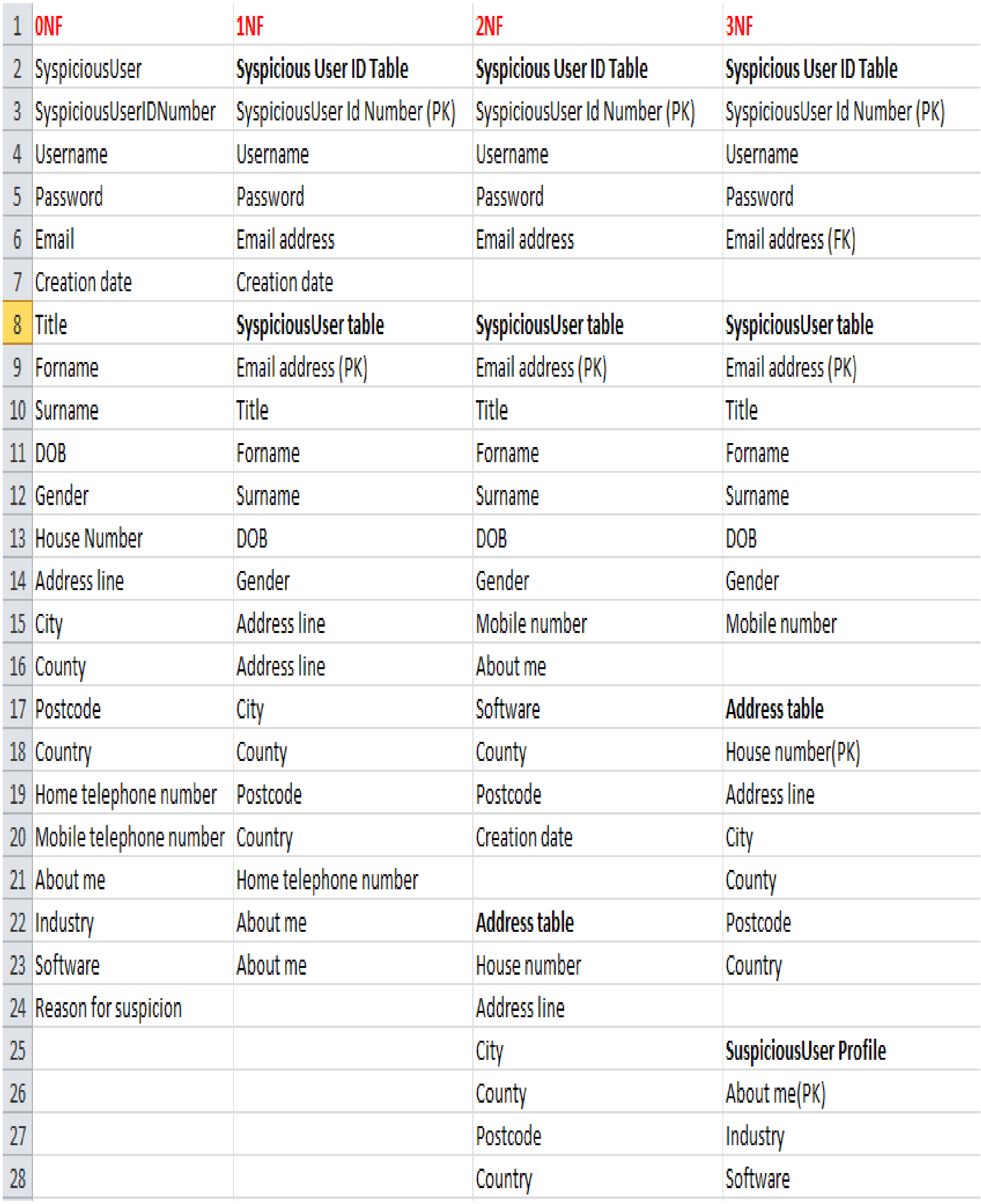
1	0NF	1NF	2NF	3NF
2	**Area of Expertise**	**Area of Expertise Table**	**Area of Expertise Table**	**Area of Expertise Table**
3	Web development	Web development(PK)	Web development (PK)	Web development (PK)
4	Web design	Web design	Web design	Web design
5	Mobile development	Mobile development	Mobile development	Mobile development
6	Microsoft development	Microsoft development	Microsoft development	Microsoft development
7	Java development	Java development	Java development	Java development
8	Title	**ScrumMasterTable**	**ScrumMasterTable**	**ScrumMasterTable**
9	Forname	Email address (PK)	Email address (PK)	Email address (PK)
10	Surname	Title	Title	Title
11	DOB	Forname	Forname	Forname
12	Gender	Surname	Surname	Surname
13	House Number	DOB	DOB	DOB
14	Address line	Gender	Gender	Gender
15	City	Address line	Mobile number	Mobile number
16	County	Address line	About me	
17	Postcode	City	Software	**Address table**
18	Country	County	County	House number(PK)
19	Home telephone number	Postcode	Postcode	Address line
20	Mobile telephone number	Country	Creation date	City
21		Home telephone number		County
22		About me	**Address table**	Postcode
23			House number	Country
24			Address line	
25			City	
26			County	
27			Postcode	
28			Country	

3.3.9 Manager



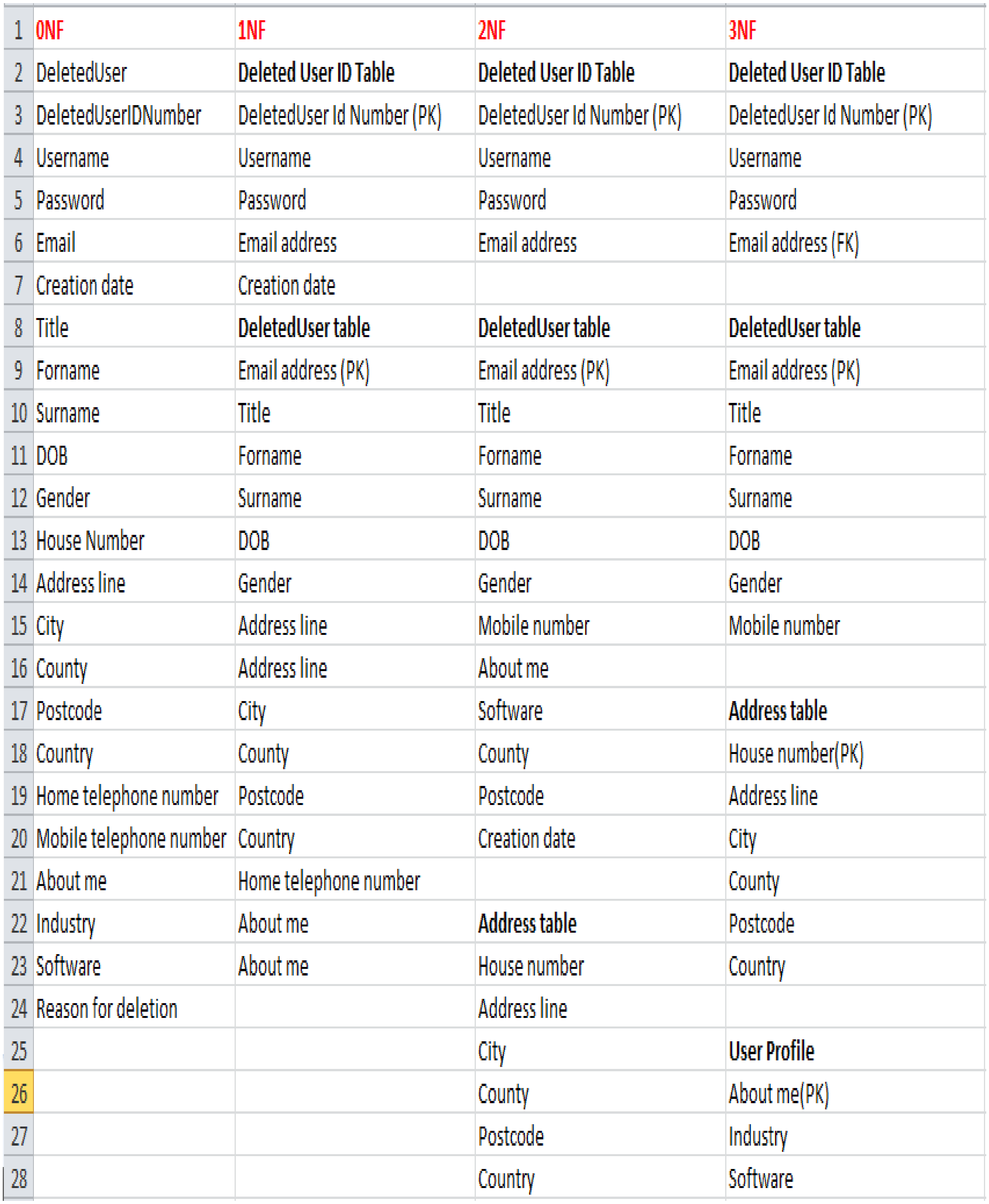
1	0NF	1NF	2NF	3NF
2	**Area of Expertise**	**Area of Expertise Table**	**Area of Expertise Table**	**Area of Expertise Table**
3	Web development	Web development(PK)	Web development (PK)	Web development (PK)
4	Web design	Web design	Web design	Web design
5	Mobile development	Mobile development	Mobile development	Mobile development
6	Microsoft development	Microsoft development	Microsoft development	Microsoft development
7	Java development	Java development	Java development	Java development
8	Title	**ManagerTable**	**ManagerTable**	**ManagerTable**
9	Forname	Email address (PK)	Email address (PK)	Email address (PK)
10	Surname	Title	Title	Title
11	DOB	Forname	Forname	Forname
12	Gender	Surname	Surname	Surname
13	House Number	DOB	DOB	DOB
14	Address line	Gender	Gender	Gender
15	City	Address line	Mobile number	Mobile number
16	County	Address line	About me	
17	Postcode	City	Software	**Address table**
18	Country	County	County	House number(PK)
19	Home telephone number	Postcode	Postcode	Address line
20	Mobile telephone number	Country	Creation date	City
21		Home telephone number		County
22		About me	**Address table**	Postcode
23			House number	Country
24			Address line	
25			City	
26			County	
27			Postcode	
28			Country	

3.3.10 Suspicious User



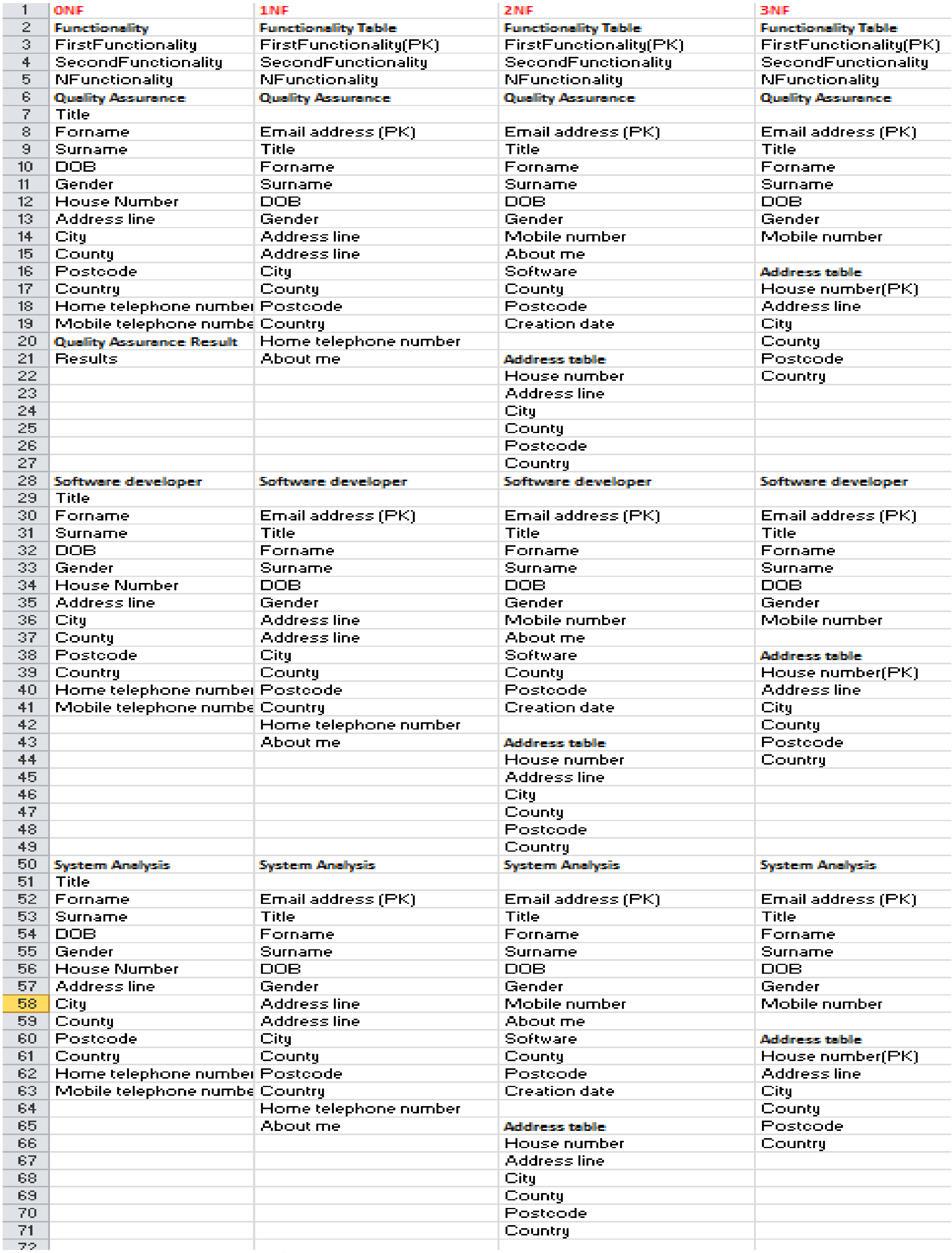
1	0NF	1NF	2NF	3NF
2	SyspiciousUser	**Syspicious User ID Table**	**Syspicious User ID Table**	**Syspicious User ID Table**
3	SyspiciousUserIDNumber	SyspiciousUser Id Number (PK)	SyspiciousUser Id Number (PK)	SyspiciousUser Id Number (PK)
4	Username	Username	Username	Username
5	Password	Password	Password	Password
6	Email	Email address	Email address	Email address (FK)
7	Creation date	Creation date		
8	Title	**SyspiciousUser table**	**SyspiciousUser table**	**SyspiciousUser table**
9	Forname	Email address (PK)	Email address (PK)	Email address (PK)
10	Surname	Title	Title	Title
11	DOB	Forname	Forname	Forname
12	Gender	Surname	Surname	Surname
13	House Number	DOB	DOB	DOB
14	Address line	Gender	Gender	Gender
15	City	Address line	Mobile number	Mobile number
16	County	Address line	About me	
17	Postcode	City	Software	**Address table**
18	Country	County	County	House number(PK)
19	Home telephone number	Postcode	Postcode	Address line
20	Mobile telephone number	Country	Creation date	City
21	About me	Home telephone number		County
22	Industry	About me	**Address table**	Postcode
23	Software	About me	House number	Country
24	Reason for suspicion		Address line	
25			City	**SuspiciousUser Profile**
26			County	About me(PK)
27			Postcode	Industry
28			Country	Software

3.3.11 Deleted Users



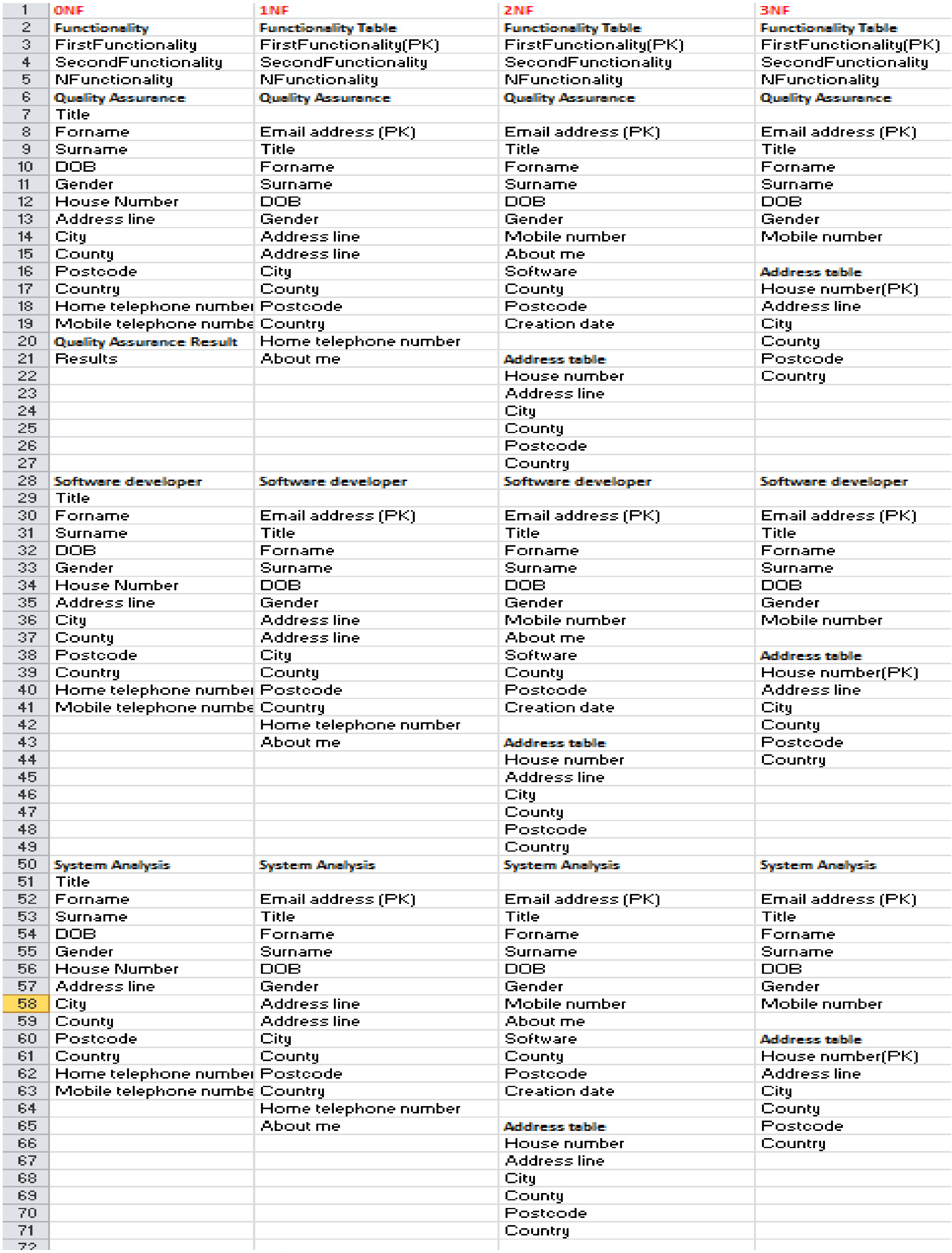
1	0NF	1NF	2NF	3NF
2	DeletedUser	**Deleted User ID Table**	**Deleted User ID Table**	**Deleted User ID Table**
3	DeletedUserIDNumber	DeletedUser Id Number (PK)	DeletedUser Id Number (PK)	DeletedUser Id Number (PK)
4	Username	Username	Username	Username
5	Password	Password	Password	Password
6	Email	Email address	Email address	Email address (FK)
7	Creation date	Creation date		
8	Title	**DeletedUser table**	**DeletedUser table**	**DeletedUser table**
9	Forname	Email address (PK)	Email address (PK)	Email address (PK)
10	Surname	Title	Title	Title
11	DOB	Forname	Forname	Forname
12	Gender	Surname	Surname	Surname
13	House Number	DOB	DOB	DOB
14	Address line	Gender	Gender	Gender
15	City	Address line	Mobile number	Mobile number
16	County	Address line	About me	
17	Postcode	City	Software	**Address table**
18	Country	County	County	House number(PK)
19	Home telephone number	Postcode	Postcode	Address line
20	Mobile telephone number	Country	Creation date	City
21	About me	Home telephone number		County
22	Industry	About me	**Address table**	Postcode
23	Software	About me	House number	Country
24	Reason for deletion		Address line	
25			City	**User Profile**
26			County	About me(PK)
27			Postcode	Industry
28			Country	Software

3.3.12 Demo App



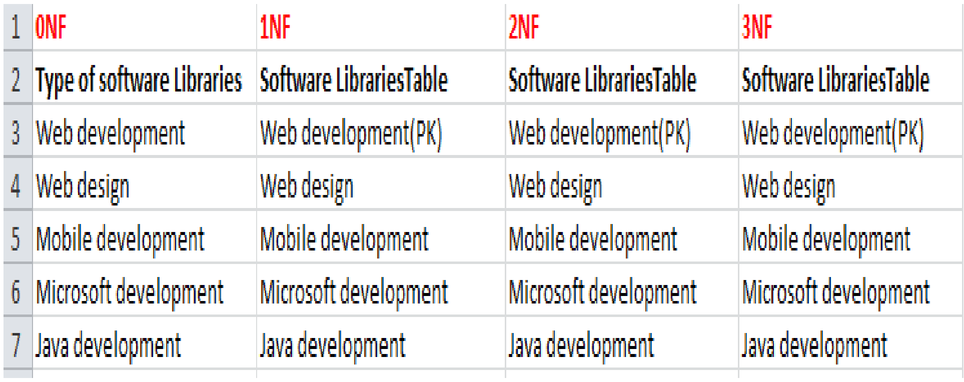
1	0NF	1NF	2NF	3NF
2	**Functionality**	**Functionality Table**	**Functionality Table**	**Functionality Table**
3	FirstFunctionality	FirstFunctionality(PK)	FirstFunctionality(PK)	FirstFunctionality(PK)
4	SecondFunctionality	SecondFunctionality	SecondFunctionality	SecondFunctionality
5	NFunctionality	NFunctionality	NFunctionality	NFunctionality
6	**Quality Assurance**	**Quality Assurance**	**Quality Assurance**	**Quality Assurance**
7	Title			
8	Forname	Email address (PK)	Email address (PK)	Email address (PK)
9	Surname	Title	Title	Title
10	DOB	Forname	Forname	Forname
11	Gender	Surname	Surname	Surname
12	House Number	DOB	DOB	DOB
13	Address line	Gender	Gender	Gender
14	City	Address line	Mobile number	Mobile number
15	County	Address line	About me	
16	Postcode	City	Software	**Address table**
17	Country	County	County	House number(PK)
18	Home telephone numbe	Postcode	Postcode	Address line
19	Mobile telephone numbe	Country	Creation date	City
20	**Quality Assurance Result**	Home telephone number		County
21	Results	About me	**Address table**	Postcode
22			House number	Country
23			Address line	
24			City	
25			County	
26			Postcode	
27			Country	
28	**Software developer**	**Software developer**	**Software developer**	**Software developer**
29	Title			
30	Forname	Email address (PK)	Email address (PK)	Email address (PK)
31	Surname	Title	Title	Title
32	DOB	Forname	Forname	Forname
33	Gender	Surname	Surname	Surname
34	House Number	DOB	DOB	DOB
35	Address line	Gender	Gender	Gender
36	City	Address line	Mobile number	Mobile number
37	County	Address line	About me	
38	Postcode	City	Software	**Address table**
39	Country	County	County	House number(PK)
40	Home telephone numbe	Postcode	Postcode	Address line
41	Mobile telephone numbe	Country	Creation date	City
42		Home telephone number		County
43		About me	**Address table**	Postcode
44			House number	Country
45			Address line	
46			City	
47			County	
48			Postcode	
49			Country	
50	**System Analysis**	**System Analysis**	**System Analysis**	**System Analysis**
51	Title			
52	Forname	Email address (PK)	Email address (PK)	Email address (PK)
53	Surname	Title	Title	Title
54	DOB	Forname	Forname	Forname
55	Gender	Surname	Surname	Surname
56	House Number	DOB	DOB	DOB
57	Address line	Gender	Gender	Gender
58	City	Address line	Mobile number	Mobile number
59	County	Address line	About me	
60	Postcode	City	Software	**Address table**
61	Country	County	County	House number(PK)
62	Home telephone numbe	Postcode	Postcode	Address line
63	Mobile telephone numbe	Country	Creation date	City
64		Home telephone number		County
65		About me	**Address table**	Postcode
66			House number	Country
67			Address line	
68			City	
69			County	
70			Postcode	
71			Country	

3.3.13 Final App



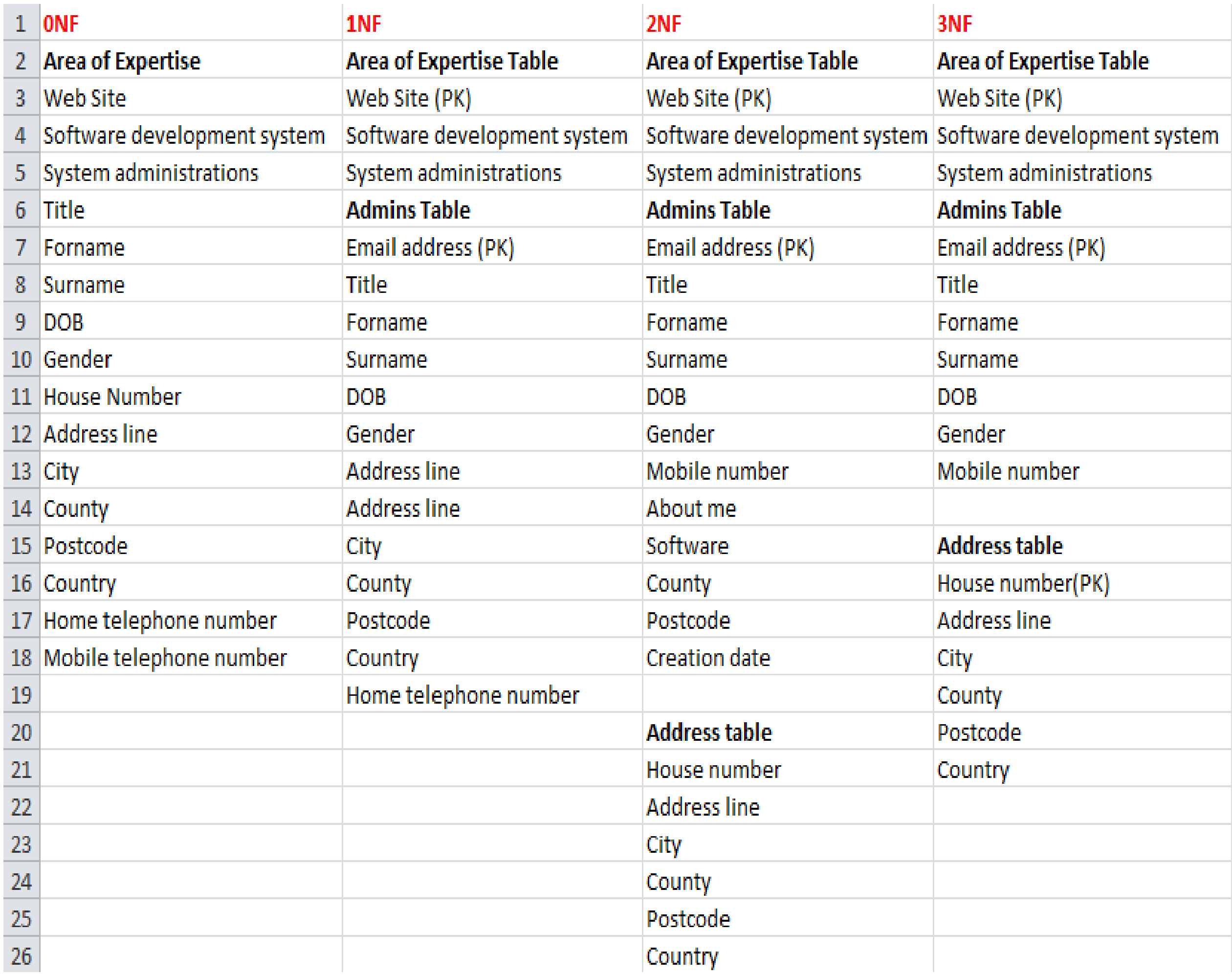
1	0NF	1NF	2NF	3NF
2	Functionality	Functionality Table	Functionality Table	Functionality Table
3	FirstFunctionality	FirstFunctionality(PK)	FirstFunctionality(PK)	FirstFunctionality(PK)
4	SecondFunctionality	SecondFunctionality	SecondFunctionality	SecondFunctionality
5	NFunctionality	NFunctionality	NFunctionality	NFunctionality
6	Quality Assurance	Quality Assurance	Quality Assurance	Quality Assurance
7	Title			
8	Forname	Email address (PK)	Email address (PK)	Email address (PK)
9	Surname	Title	Title	Title
10	DOB	Forname	Forname	Forname
11	Gender	Surname	Surname	Surname
12	House Number	DOB	DOB	DOB
13	Address line	Gender	Gender	Gender
14	City	Address line	Mobile number	Mobile number
15	County	Address line	About me	
16	Postcode	City	Software	Address table
17	Country	County	County	House number(PK)
18	Home telephone numbei	Postcode	Postcode	Address line
19	Mobile telephone numbe	Country	Creation date	City
20	Quality Assurance Result	Home telephone number		County
21	Results	About me	Address table	Postcode
22			House number	Country
23			Address line	
24			City	
25			County	
26			Postcode	
27			Country	
28	Software developer	Software developer	Software developer	Software developer
29	Title			
30	Forname	Email address (PK)	Email address (PK)	Email address (PK)
31	Surname	Title	Title	Title
32	DOB	Forname	Forname	Forname
33	Gender	Surname	Surname	Surname
34	House Number	DOB	DOB	DOB
35	Address line	Gender	Gender	Gender
36	City	Address line	Mobile number	Mobile number
37	County	Address line	About me	
38	Postcode	City	Software	Address table
39	Country	County	County	House number(PK)
40	Home telephone numbei	Postcode	Postcode	Address line
41	Mobile telephone numbe	Country	Creation date	City
42		Home telephone number		County
43		About me	Address table	Postcode
44			House number	Country
45			Address line	
46			City	
47			County	
48			Postcode	
49			Country	
50	System Analysis	System Analysis	System Analysis	System Analysis
51	Title			
52	Forname	Email address (PK)	Email address (PK)	Email address (PK)
53	Surname	Title	Title	Title
54	DOB	Forname	Forname	Forname
55	Gender	Surname	Surname	Surname
56	House Number	DOB	DOB	DOB
57	Address line	Gender	Gender	Gender
58	City	Address line	Mobile number	Mobile number
59	County	Address line	About me	
60	Postcode	City	Software	Address table
61	Country	County	County	House number(PK)
62	Home telephone numbei	Postcode	Postcode	Address line
63	Mobile telephone numbe	Country	Creation date	City
64		Home telephone number		County
65		About me	Address table	Postcode
66			House number	Country
67			Address line	
68			City	
69			County	
70			Postcode	
71			Country	
72				

3.3.14 Software Libraries



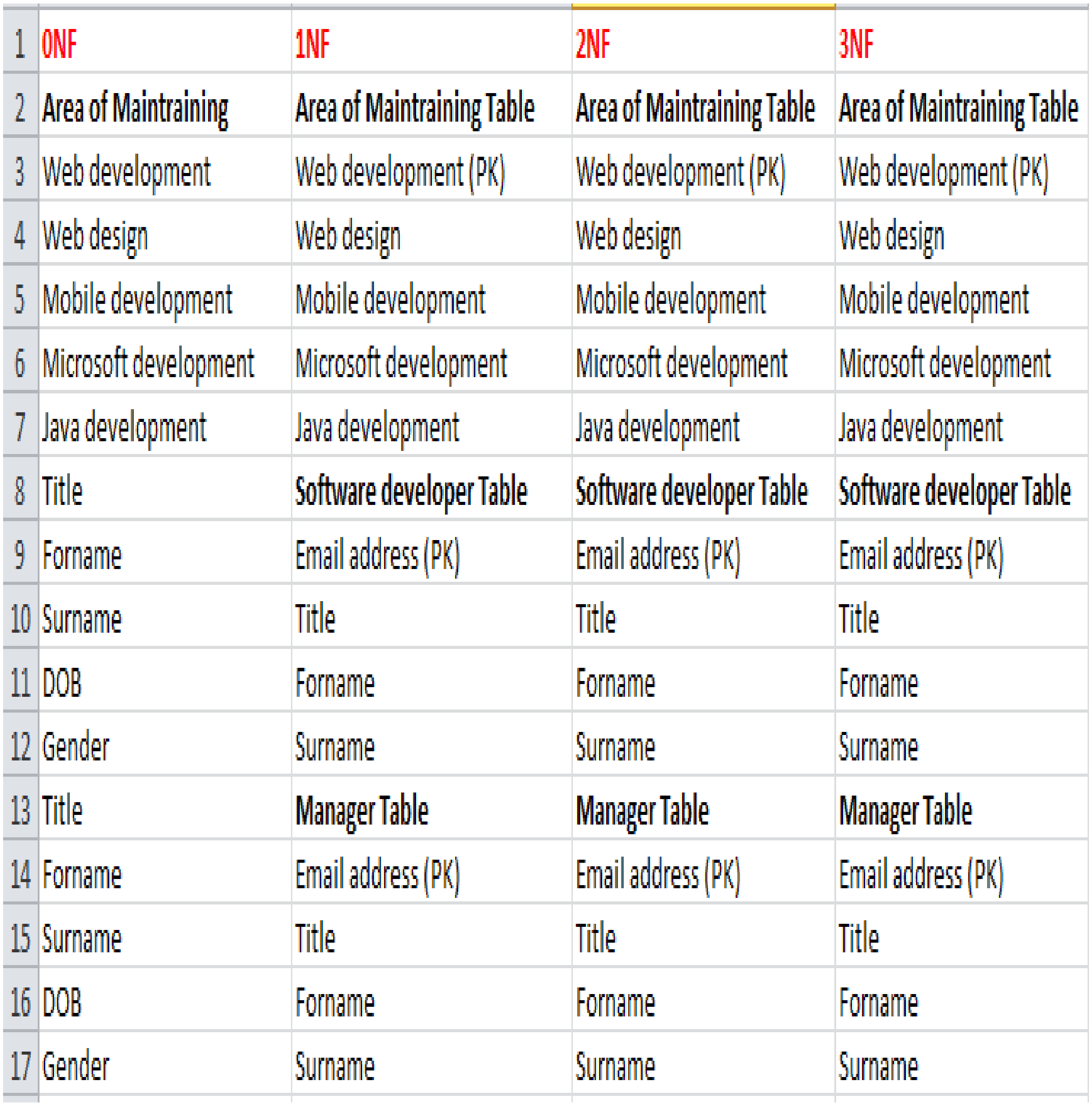
1	0NF	1NF	2NF	3NF
2	**Type of software Libraries**	**Software LibrariesTable**	**Software LibrariesTable**	**Software LibrariesTable**
3	Web development	Web development(PK)	Web development(PK)	Web development(PK)
4	Web design	Web design	Web design	Web design
5	Mobile development	Mobile development	Mobile development	Mobile development
6	Microsoft development	Microsoft development	Microsoft development	Microsoft development
7	Java development	Java development	Java development	Java development

3.3.15 Admins



1	0NF	1NF	2NF	3NF
2	**Area of Expertise**	**Area of Expertise Table**	**Area of Expertise Table**	**Area of Expertise Table**
3	Web Site	Web Site (PK)	Web Site (PK)	Web Site (PK)
4	Software development system	Software development system	Software development system	Software development system
5	System administrations	System administrations	System administrations	System administrations
6	Title	**Admins Table**	**Admins Table**	**Admins Table**
7	Forname	Email address (PK)	Email address (PK)	Email address (PK)
8	Surname	Title	Title	Title
9	DOB	Forname	Forname	Forname
10	Gender	Surname	Surname	Surname
11	House Number	DOB	DOB	DOB
12	Address line	Gender	Gender	Gender
13	City	Address line	Mobile number	Mobile number
14	County	Address line	About me	
15	Postcode	City	Software	**Address table**
16	Country	County	County	House number(PK)
17	Home telephone number	Postcode	Postcode	Address line
18	Mobile telephone number	Country	Creation date	City
19		Home telephone number		County
20			**Address table**	Postcode
21			House number	Country
22			Address line	
23			City	
24			County	
25			Postcode	
26			Country	

3.3.16 Maintain App



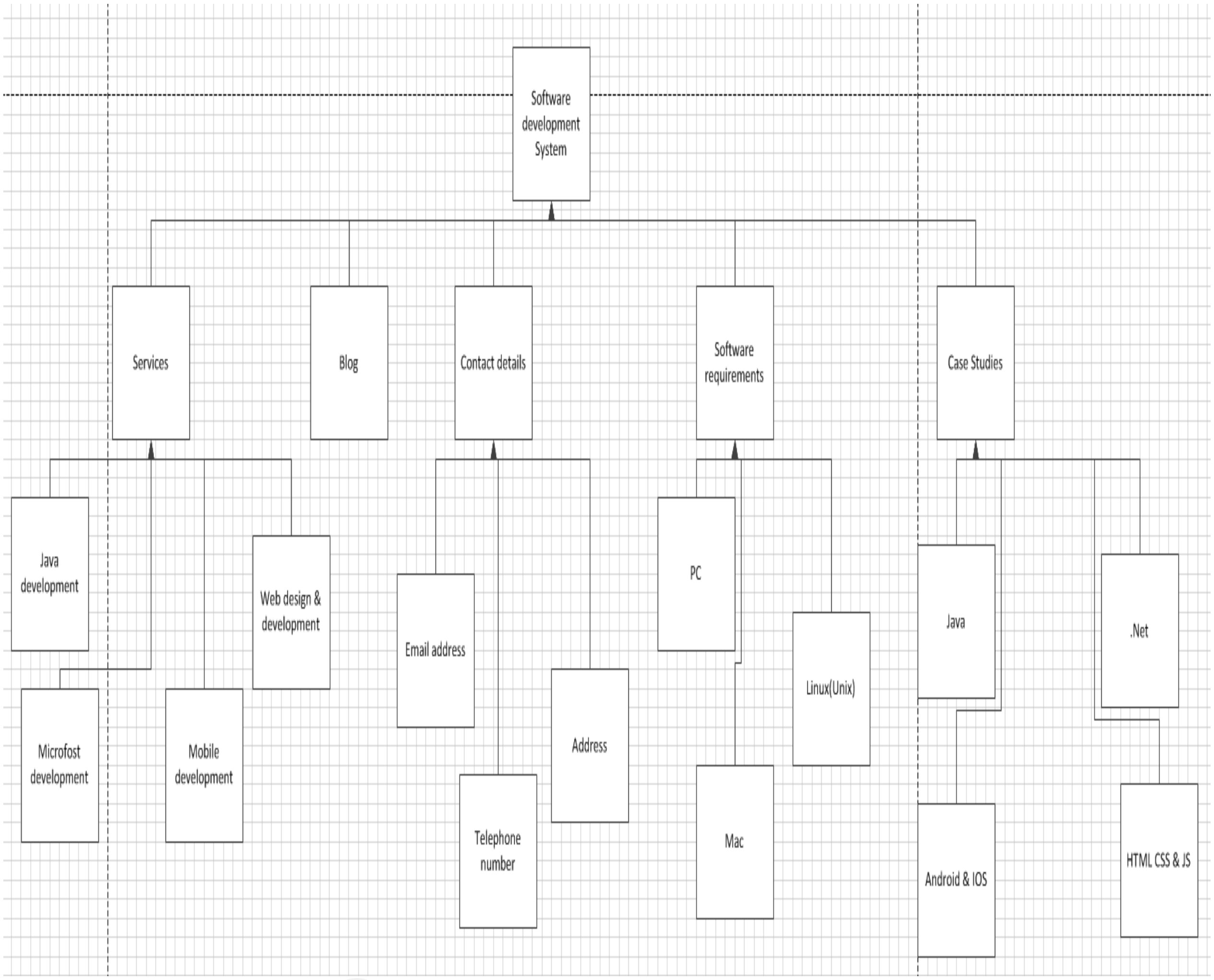
1	0NF	1NF	2NF	3NF
2	**Area of Maintraining**	**Area of Maintraining Table**	**Area of Maintraining Table**	**Area of Maintraining Table**
3	Web development	Web development (PK)	Web development (PK)	Web development (PK)
4	Web design	Web design	Web design	Web design
5	Mobile development	Mobile development	Mobile development	Mobile development
6	Microsoft development	Microsoft development	Microsoft development	Microsoft development
7	Java development	Java development	Java development	Java development
8	Title	**Software developer Table**	**Software developer Table**	**Software developer Table**
9	Forname	Email address (PK)	Email address (PK)	Email address (PK)
10	Surname	Title	Title	Title
11	DOB	Forname	Forname	Forname
12	Gender	Surname	Surname	Surname
13	Title	**Manager Table**	**Manager Table**	**Manager Table**
14	Forname	Email address (PK)	Email address (PK)	Email address (PK)
15	Surname	Title	Title	Title
16	DOB	Forname	Forname	Forname
17	Gender	Surname	Surname	Surname

4. Web Design

As mentioned in the introduction section, web methodology will be used for the creation of the website. The website has already been planned and analysed within the documentation. This section will include making a content model, making a site structure, deciding the visual style of the website, including fonts, colour schemes, implementing a sample master page and so on. It will be necessary to figure out what types of navigation will be used and where to put them, what the pages are going to look like and the logo designs, accompanied by the finished logo.

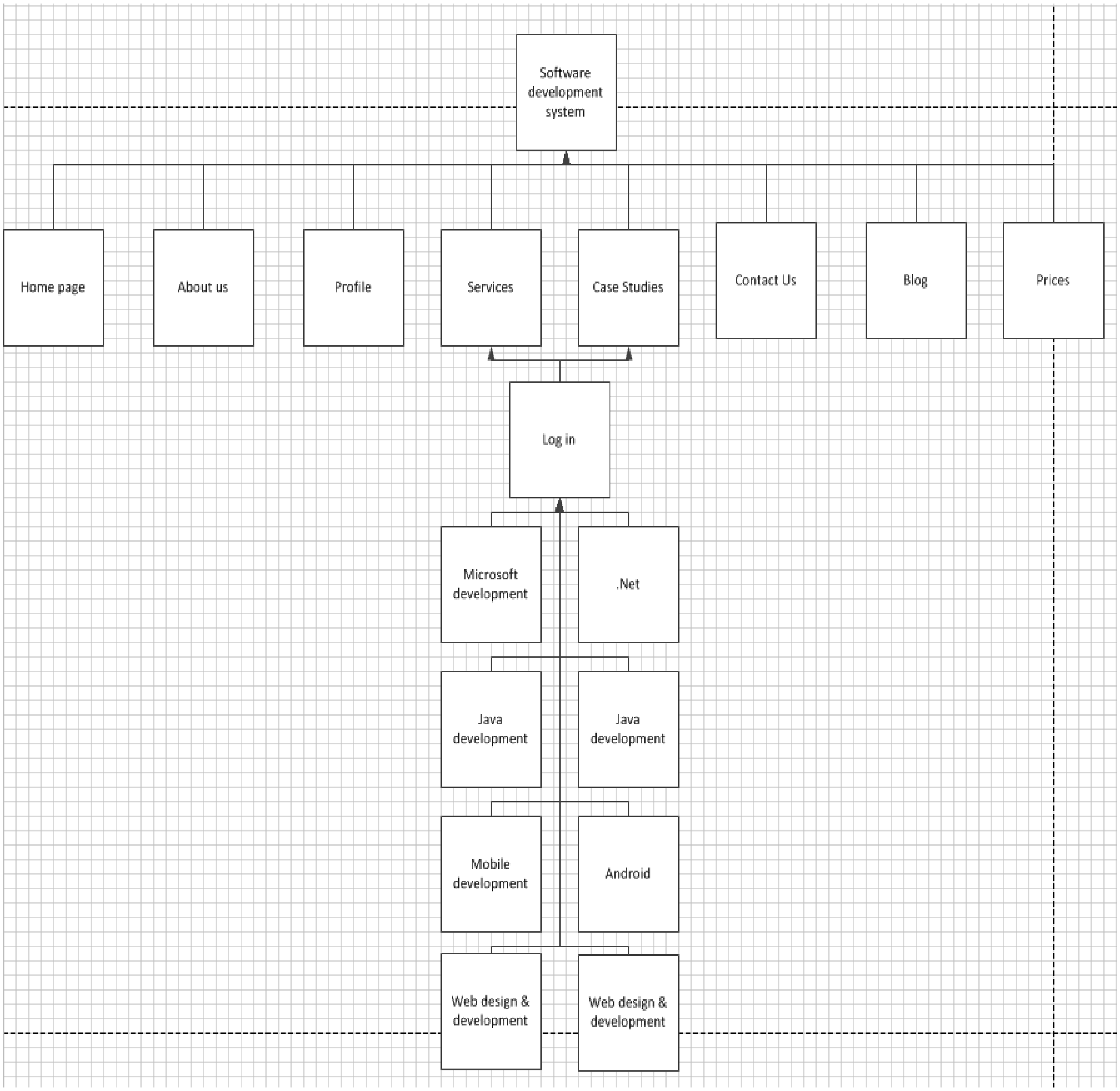
4.1Content Model

The content model that was build is a foundation for the site structure. The site structure takes the information from the content model and creates pages for the website. Similar headings will be used for the page titles in the site structure.



4.2 Site structure

In order for the structure to be successful, it will need to organise it in a way the users will understand. The site structure will link closely to the content model. The content model will be brought down in to pages in order to create the site structure. As mentioned in the requirements, a non-registered user can only see limited pages. The registered users have the ability to log in to the website and have full access to all the facilities (All the services and case studies of the system). The site structure below would be implemented on to the website.



4.2 Visual style

The visual style for the website will closely link to how the pages will look. As a master page is being created, the colours will stay the same as the user navigates through the website. The consistency of colours gives the website a professional feel; this is why it has been decided to use a master page.

4.2.1 Fonts

The main font that will be used for the content will be Verdana, Serif, and font size 11px. It has been decided to use this font because it is easy to read no matter what background is in place. The headings for the website will also be Verdana, Serif, but will be in font size 12 or more depending on the heading and bold.

4.2.2 Colour Scheme

The colour scheme will feature a black background, with a light grey footer. The other main colours to feature will be white, black, red, purple, green and silver. The hexadecimal web colour codes are as follows:

Purple	Dark Grey	Light Grey	White	Black	Red	Silver	Green

- Purple - #800080; Dark Grey - #333333
- Light Grey - #CCCCCC; White - #FFFFFF
- Black - #000000; Red – #FF0000
- Silver - #CCCCFF; Green - #008000

4.2.3 Background

The background will be a plain colour background following the colour scheme. The colour that has been chosen as the background is black and the hexadecimal code for black is #171612. I have chosen black as it is a pretty strong colour, this means I can make my texts, images and logos much more eye-catchy on this black background.

4.2.4 Video and audio

When Audio and video are being used within the website, they will have to be set in appropriate formats. The videos will mainly be posted by the website administrator depending on the pages and what content it holds. Users will also be able to upload their videos on the websites forum. The website will accept different Videos like MPEG4, AVI, MKV and Mov. The audio will normally follow with the video, however if a user wants to upload separate audio, the preferred format would be MP3 or WAV.

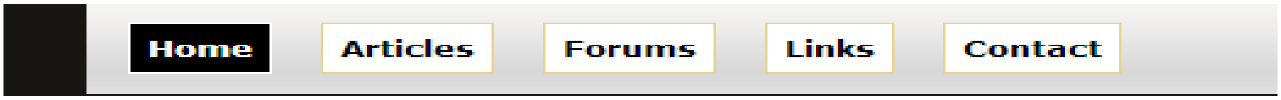
4.2.5 Screen size

The lowest screen size that is going to be used is 4.2.6 Graphics (including icons, rollovers and Logos 800 x 600 and the highest will be 1280 x 1024. The in-between average will be, 1024 x 768 and will be the most commonly used due to users using this resolution. The design of the website is going to use the lowest resolution. Users wanting a bigger screen size and resolution will get it. 800 x 600 won't have up to date browser.

4.3 Navigation

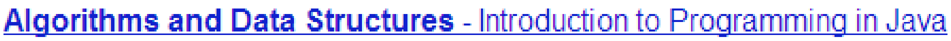
It has been decided that there are going to be several types of navigation within the website to add an extra ease of accessibility for users who find it difficult using websites. The 2 pieces of navigation that will be used are Hyperlinks and Macro buttons.

The next bit of navigation will be the buttons. Buttons are relatively simple to program and will use a basic macro application. The button (once clicked) will redirect the user to the selected page. Buttons are easy to identify and are normally associated with navigation. Here is a visual aid of the buttons, according to the navigation scheme. These buttons are good but could be improved by a designer.



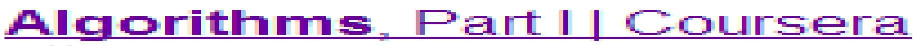
The final piece of navigation to be used within the website is hyperlinks. The hyperlinks will be used for the 'in text type of navigation. They will be easy to access and find due to the colour and underlined state of the key word (blue for un-used and purple for already used standardized). The hyperlink (once clicked) will re-direct the user to a similar page to the word clicked. It may be that the hyperlinks will redirect the user away from the website and to an external website. When this happens it will be made sure this was feasible and legally correct. The user would be informed that they were leaving the website via an external window. Here are an example of an unused hyperlink and a used hyperlink.

Un-used – The link remains blue and unused



Algorithms and Data Structures - Introduction to Programming in Java

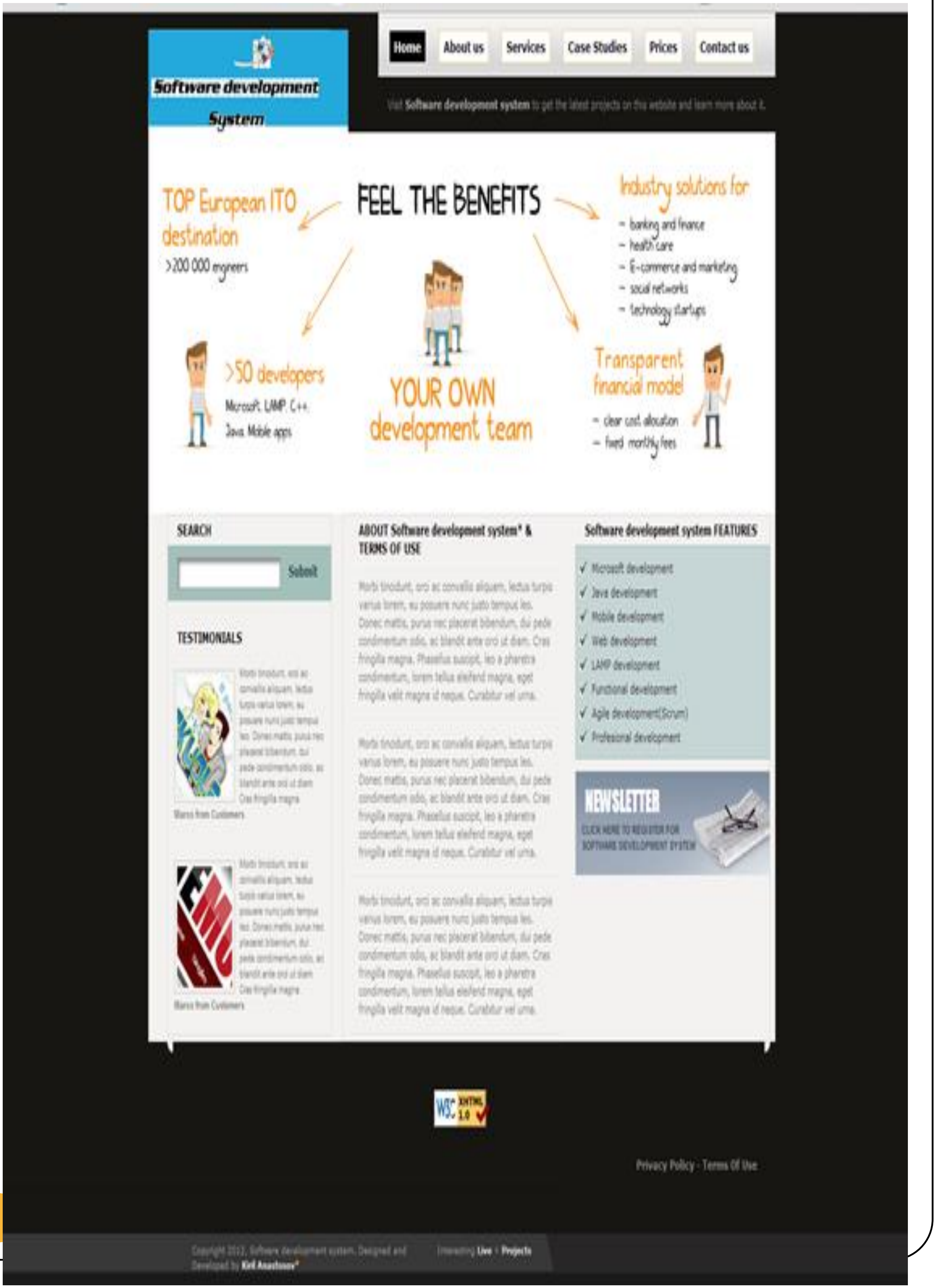
Used – The link has turned purple, indicating it has already been accessed



Algorithms, Part I | Coursera

4.4 Pages

As the website will be developed for an online software development, pages for the users have to be designed to view and navigate around. To make this process easier, it was decided that it is going to be designed a master page, so the other pages will look the same, minus the content inputted in. This master page will give an idea of how the whole website will look and it will strictly follow the visual style. This is an example of what has been done so far. I have done this myself, the design and the development. The website is in process of development but this will be the main aspect with a few improvements.



Software development System

Visit **Software development system** to get the latest projects on this website and learn more about it.

FEEL THE BENEFITS

TOP European ITO destination
>200 000 engineers

Industry solutions for
- banking and finance
- health care
- E-commerce and marketing
- social networks
- technology startups

>50 developers
Microsoft, LAMP, C++,
Java, Mobile apps

YOUR OWN development team

Transparent financial model
- clear cost allocation
- fixed monthly fees

SEARCH

Submit

TESTIMONIALS

Morbi tincidunt, orci ac convallis aliquam, lectus turpis varius lorem, eu posuere nunc justo tempus leo. Donec mattis, purus nec placerat bibendum, dui pede condimentum odio, ac blandit ante orci ut diam. Cras fringilla magna

Marco from Customers

Morbi tincidunt, orci ac convallis aliquam, lectus turpis varius lorem, eu posuere nunc justo tempus leo. Donec mattis, purus nec placerat bibendum, dui pede condimentum odio, ac blandit ante orci ut diam. Cras fringilla magna

Marco from Customers

ABOUT Software development system* & TERMS OF USE

Morbi tincidunt, orci ac convallis aliquam, lectus turpis varius lorem, eu posuere nunc justo tempus leo. Donec mattis, purus nec placerat bibendum, dui pede condimentum odio, ac blandit ante orci ut diam. Cras fringilla magna. Phasellus suscipit, leo a pharetra condimentum, lorem tellus eleifend magna, eget fringilla velit magna id neque. Curabitur vel urna.

Morbi tincidunt, orci ac convallis aliquam, lectus turpis varius lorem, eu posuere nunc justo tempus leo. Donec mattis, purus nec placerat bibendum, dui pede condimentum odio, ac blandit ante orci ut diam. Cras fringilla magna. Phasellus suscipit, leo a pharetra condimentum, lorem tellus eleifend magna, eget fringilla velit magna id neque. Curabitur vel urna.

Morbi tincidunt, orci ac convallis aliquam, lectus turpis varius lorem, eu posuere nunc justo tempus leo. Donec mattis, purus nec placerat bibendum, dui pede condimentum odio, ac blandit ante orci ut diam. Cras fringilla magna. Phasellus suscipit, leo a pharetra condimentum, lorem tellus eleifend magna, eget fringilla velit magna id neque. Curabitur vel urna.

Software development system FEATURES

- ✓ Microsoft development
- ✓ Java development
- ✓ Mobile development
- ✓ Web development
- ✓ LAMP development
- ✓ Functional development
- ✓ Agile development(Scrum)
- ✓ Profesional development

NEWSLETTER
CLICK HERE TO REGISTER FOR SOFTWARE DEVELOPMENT SYSTEM

Privacy Policy - Terms Of Use

Copyright 2012, Software development system. Designed and Developed by **Kiril Anastasov***

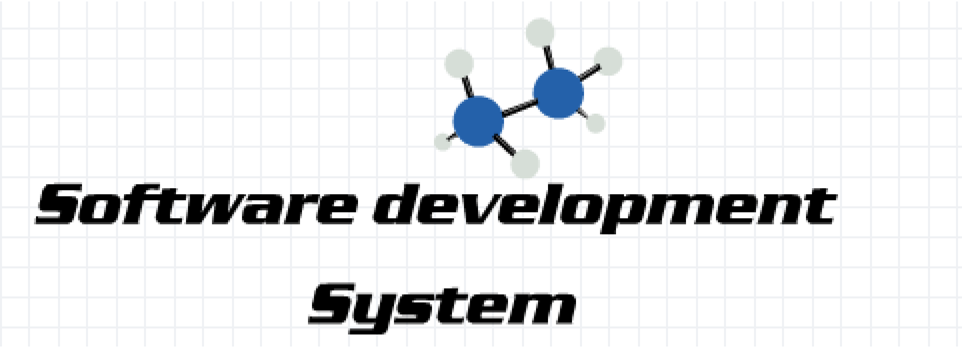
Interesting **Live** : **Projects**

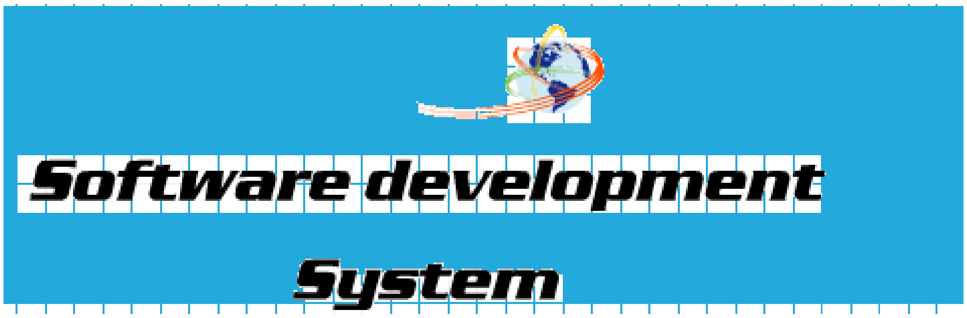
4.4.1 Response time

The website will be looking to hit the target of 0.5 second for the response time between pages. The ideal response time is between 0.1 and 0.3 second however seen as though the website will be featuring images, logos and videos, the more realistic response time is between 0.3 and 0.7 seconds. The user may notice a delay in response time but it shouldn't come anywhere near close to the 3 second limit. The target is reasonable and manageable for this system. The response time is very important because if the user is not familiar with the system this will be his first interaction and the first impression is vital.

4.5 Logo Design

As the system is being created from scratch, a logo is needed to be designed for the system. Logo maker is going to be used to easy the design of the logo. In this way the logo is going to be professional and it is going to save time and nerves.





Both logos seem good considering the previous experience with logos. So the second one is going to be used in the system because the idea and how it looks like. Positive expectations are predicted about the logo, because several people have provided a positive feedback.

5. Critical review

5.1 Critical review of ICA criteria or deliverables

5.1.1 C1 – Introduction

The introduction for the project seemed the strongest section. It gives the reader a deep understanding of what the project is and what it is going to do. The introduction is split in to 7 sections. Normally introductions are shorter than the one in this document and not as clear, but the purpose is to explain better what the software development system was all about, what methodologies were used and so on. The first sub section introduced what the project was. It was very important to get this out there first thing. If this was placed in a later section, the reader might not know what he project was about. The software development system has been explained sufficiently in this sub heading, without too much unnecessary details and confusion. The next section explains the planning and scheduling of the work to do before the deadline (19th April 2013). It was decided that the best way to do this was to clearly lay out Teesside university holidays in writing and create a gannt chart for more of a visual representation and guidance. The gannt chart was created in Microsoft project and it gives the reader a better understanding of when sections of the work were going to be completed and handed in. It was chosen to research in to the methodologies available within the introduction section so it shows that it has given a lot of thought early on to see which will be used. The advantages and disadvantages have been provided for personal gain to see which methodology is best for this project and to make it clearer for the people reading it. A better understanding of the methodologies have been gained and decided to choose one after seeing what the others have to offer. The agile method was by far the best choice with the vast amount of advantages and least disadvantages. Next, a small section on UML and web design has been done. It is explained what each of them were and how it was going to be implement within the project. Again it was thought that this was valid as it is letting the reader know more about my project by explaining the terms. This will hopefully show that a great deal of research has gone in to this project, and it will show that the project has not been done it in the last week before the deadline. Finally, it has been shown a data flow diagram key. This explains what the symbols are within my data flow diagrams and context diagrams. It has been decided to put this key in the introduction instead of later in the project so it could be understood clearly how to create the diagrams later on. The introduction is a relatively bigger than normal but this way it was given a head start researching in to the harder tasks later on in the project and the other readers of the document can benefit as well.

5.1.2 C2 – Systems Analysis: Project Deliverables

The system analysis section is going to be reviewed in to two parts, in this section it will review the terms of reference. The terms of reference includes the statement of purpose, project requirements, target audience and restrictions and constraints. It was important that it was explained these sections in detail to kick start the project. The statement of purpose explains the purpose of the system briefly. This could have been improved on and extended but it was unsure to be done so as it was feared that it will talk too much. This purpose is clean and straight to the point which the readers will appreciate instead of ploughing through lots of text.

The user requirements show all the possible scenarios the system has to deal with. Again the requirements have been laid out in bullet point format, so it is easy for the reader to understand. Presentation is vital when creating reports of this standard as you don't want to blast the reader with unlimited and never ending text. The requirements were modified throughout the report so it was necessary to continuously keep changing the project. This section was one of the strongest as it give something to work on for the rest of the project as it was known what the system was all about.

The target audience is always tricky because the system can be used by anyone and anyone can appreciate it. However, as it has been built by a software developer, it is easier to understand what type of audiences this type of system will attract. This helped to make a prediction what age category it will be aiming for in order to interest users to the system. The reason that t this project was chosen is because it is done by keen developer, this gave an edge in analysis and designing the system as it is known what to look for in professional looking applications. If it was necessary to create another report it would be followed what the person doing it is interested in as it makes you get more involved with your work.

Finally it was necessary to pick out the restrictions and constraints. This was a fairly simple category as they were laid out in front. These will be very similar with all reports as they all have time deadlines, certain types of machines they are and aren't allowed to use and so on. There was not much that could have been done to improve this section due to this main factor.

5.1.3 C2 – Systems analysis: Demonstrate Diagrams

In this section of the systems analysis, it is going to be reviewed the different diagrams that have been produced. The review will be split in to 2 sections. The first section will review both the top level data flow diagram and the lower level diagrams. The second section will be designated to reviewing the 5 different UML diagrams. What is missed in the data flow diagrams is aimed to make up in the UML diagrams.

The top level data flow diagram shows an exploded view of the context diagram. At first constructing the top level diagram seemed a great task, but after researching how to construct them correctly, it started to come together. It was used the fragment procedure where it was grabbed the event list and turned them in to processes. After labelling the processes it was necessary to link them to their terminators as well as their inputs and outputs. After that the diagram really came together and my confidence grew. The data stores came naturally and my top level data flow diagram was made. If it was necessary to design another top level diagram it would record these fragments separately and include them in the report. However, with the time constraints involved with the project, it could not be recorded. It was used Microsoft Visio instead of ASCENT, due to software restriction. ASCENT would have been better as you can explode each process individually. Microsoft Visio proved hard to work with, as a lot of formatting was to be done in order to get the aesthetically pleasing look to satisfy the diagram needs. However, after a lot of work and many hours later, the diagrams look very appealing due to the formatting. The lower level diagrams took longer to create. Each process was taken and exploded them further in order to explain the system. The diagrams have been drawn and a paragraph was given explanation just in case the reader doesn't understand any part of the lower level data flow diagram. Having this paragraph ideal just in case the reader is a non IT user, this will make it much easier to understand.

To add more detail to the project, it was decided to create some UML diagrams. Hopefully the UML diagrams will counterbalance any weaknesses the Data flow diagrams hold. For the activity diagrams it was chosen 5 processes (log in, profile update, system requirements, demo application and final application) to go in to more detail, as these processes could be explained further. It has been tried to explain them as best as it could have been done with the knowledge and the power of UML. However, they are professional looking. It was also created a use case diagram to show the interactions between the users of the system. This model is a brief outline of what the users have to do in the system. The use of the generalisation notations was necessary and they reduced the size of the diagram significantly (Actor generalisation). The class and state diagrams were linked together. The class diagram showed the structure of the system and would contribute the entity relationship diagrams later in the project. It was thought that it was necessary to include these diagrams to add more detail to the report. The reason it has only one state diagram is because the behaviour of the other classes in the diagram did not show much behavioural difference, where the application development class could be explore further.

5.1.4 C3 – Systems Design

The design stage of the report could not be started until it was completed the data flow diagrams. The design stage was split up in to 4 parts; Data stores – structures and elements, Process descriptors, Entity relationship diagram and normalisation. First it has to list each data store and explain the different structures and elements the data store contained. This would be the starting point to building a database. As this has not been done before, it was tricky at first. But after breaking several structures down in to elements it seemed to get the idea. It was difficult when it came to the element formatting as you may see and this sometimes proved difficult. A lot of the elements had to leave formatting blank. However, some of the elements did not need formatting, therefore so they were left as that. This process gave me a much better understanding of what part data stores play in the data flow diagrams and what data was actually outputted and inputted in to them. If it is to be done again in a future project it would have the knowledge of how to do them, but it would be taken the time and effort to fill each formatting space for each element.

The process descriptors were the easiest out of the four design sections in my opinion. The rules for describing the processes (from the top level data flow diagram) were simple. It had to be described each process in one or two lines, label the inputs and outputs (data flow) which occur in the top level data flow diagram and write a mini spec about them (what the actually do in the system). This is where it was found a bit challenging. It was chosen to write a paragraph about what the process does and attempt to include an algorithm. This is where the process descriptors could have been improved. Again, in the future it would be dedicated more time to improving these algorithms to improve the project.

Next It had to be created an Entity relationship diagram. There was experience with ERDs in my Database module so understanding how to do them was not a problem. However, it should have been attempted to use all the data stores in both the top and lower level data flow diagrams but it has only been used the top level data stores. It was thought splitting up the unregistered user and the registered user was a good idea as they are both different actors and both having different responsibilities. This does not mean to say the diagram was split. The ERD shows how the system works on a whole and what tables are interacted with by which user by using the crow's feet method to show the one too many relationships. It has not been included any many to many relationships due to the unprofessional attribute these relationships bring to the table. Where the many too many relationships would have formed, it was added an extra table to form one to many. This way the diagram is easier to read. Again doing this ERD diagram taught me a lot and how important the data stores role is in this whole project.

Finally, it had to be created a bottom down approach and create the tables using normalisation. Scepticism could be seen at first when creating the tables using normalisations but the method gets rid of redundant data, therefore making the tables much easier to understand. It took the data from the data stores structures and elements and normalised it in to 3rd normal form. This proved tough as it has not been done in previous modules or at all. However, it goes the hang of it and (even if there are a few mistakes) they were completed. It was found this difficult and the end result may not be to the best standard but it gives the reader a good understanding of what my tables will look like and how it would like them implemented.

5.1.5 C4 – Web Design

Finally we come to the web design. The design of the website was chosen to use the simple web methodology. Under the design section of SWM there are certain headings to follow. It was made sure to strictly follow these headings, as well as adding additional ones where necessary. The headings were: content model, site structure, visual style, pages and logo design.

First off The content model needed to be done. The content model shows the reader what sort of information is going in to the website. It was used a hierarchy model to demonstrate this as it is easy to decipher which bits of information goes where instead of having a large piece of text to read through. The content model that will go on to be built the site map (architecture). The site map again uses a hierarchy diagram to deliver its message. This shows what headings it will use and which page lives within each heading. Again, the use of the diagram eases the user in to the design and understands how it will look much clearer. It was thought that the use of these 2 diagrams will give the documentation a change in style and will give the user a break from reading massive chunks of text.

The next heading went in to detail about the visual style of the website. It was chosen to focus on 4 main sections (font, colour scheme, and background and screen resolution). It was important that it was written a small paragraph for each of these to explain what the website will look like. The colour scheme is good where it shows which colours will be used. It was decided to use filled text boxes with the chosen colour instead of text to give a more visual aid towards the section, this makes the report stand out a bit more and again gives the reader a break from continuous text reading. The next section explained what the pages look like. For this it was used a storyboards. Instead of using lots of different storyboards I designed a master page. The master page will be a structure for all the other web pages stated in the site map. If it was to develop a storyboard for each page of the website, the time limit will not be followed. The master page seemed like the best choice and proved evident. It has been labelled where the content place holders will be within the master page. These content place holders means that everything inside this box is editable, therefore the other pages can be formed.

5.2 Critical review of ICA

5.2.1 ICA goals

The ICA goals were very simple. It had to be created a piece of documentation which covers the analysis and design phase of a project. The main aim was to create the documentation, so an outside user could easy understand what is happening in each section and could go on and implement the system proposed, from the document. It is thought the headings within the documentation break down the project in to manageable chunks, steadily working its way through the analysis and design stages. Deciding to split up the project in to these many headings came easy, as it has been done so in previous reports and as it is the norm for report writing. A user (who has had no interaction with the system before) could come along, read the analysis and design and go on to implement it. The main reason behind this is because of the straight forward approach that has been provided.

The project scope and requirements get straight to the point without the reader having to navigate through the many unnecessary words. The context diagram is a bit crammed in, but it was decided to keep this diagram instead of splitting it up in to several diagrams. This has been done because it has gone in to much more detail in the top level data flow diagrams and the lower level data flow diagram. The design is the most complicated of the bunch and took me longer to create (as all design stages do) as it was created for the first time.

Finally the web development was relatively easy section, due to my web development knowledge and experience. Using the simple web methodology was the best option due to its ease of use. The document has achieved it's main goal due to these points, however it could have been improved by actually setting more goals but it was thought that this was the best one because there was not enough time to implementing the system myself.

5.2.2 ICA achievements

The ICA achieved all of its goals in terms of creating a document that an outside user could understand. It was important that it got this across in my analysis phase before moving on to the design. This ICA is ready for implementation and a system could be created from this report. A few gaps may be present in the ICA due to the lack of experience it have had when designing systems. These gaps may occur in the data stores – structures and elements section and the normalisation section. However, the project could be implemented by someone other than me. This was my main goal and it was achieved.

5.2.3 Not Achieved

The main concern, when finalising the report was leaving the normalization process with minor errors. Even though the main concept was correct and efficient there were some parts that could have been improved and corrected. The normalization proved to be hard because it has not been done before.

5.2.4 Evaluation

As a whole the ICA is pleasant and this report could be used to implement the system. The data flow diagrams show the system in a lot of detail and help build a better understanding of what it want the system to do. The lower level diagrams show the system in much more detail, however, if it was more time, it would be gone in to even more detail with some of the lower level processes, as they could have been exploded even further. If the project is to be done over, it would certainly be consider going in to more details. This would probably benefit me when a report for my final year project is being done. It would have also been tried harder to master the ASCENT. However, it is easily frustrated with the application and resorted to Microsoft Visio. Using Microsoft Visio to create diagrams wasn't the easiest of tasks, but after a lot of work, it worked and they look aesthetically pleasing. In the future, it would put aside a few days to master ASCENT, and build my knowledge further as it would have made creating the data flow diagrams a lot easier.

5.2.5 Reflection

The development of the ICA was tough, but in the end the results are pleasing. The document splits up in to several sections making the report much easier to understand. If someone were to implement my system, certain parts would be easy to identify using the contents page. The main parts are the Analysis phase and web design. These were taken from the agile (Scrum) system which was described in the introduction. The data flow diagrams in the analysis phase are the strongest section. The lower level diagrams go in to more detail than expected and it was the time that helped me. As each process was explored from the top level, the extra lower processes formed and them lower processes would give a much better understanding of the system.

5.2.6 Personal development

As this ICA has been developed, a lot of new skills were learned. The main improvement includes drawing draw data flow diagrams. Before this ICA, data flow diagrams were not the strong point, but after a lot of self-research and lectures, it was found out how to create them (using the data fragments method). For future projects the necessary diagrams could be done due to this module and this will help me in the final year project.

A good perspective on time management has been gained. Before this project everything used to be left until the last minute. But this module has taught me that you need to spread the work over a long period of time (in this case the whole year). This type of project could not be done the day before the deadline. It was finished more than 10 days earlier. This gives me more time to work on my other assessments. This new found time management skill will benefit me greatly when it comes to do my final year project as the tools to success will be present.

A good understanding of designing a system has been gained and what parts need to be done. Before this module began there was scepticism of what needed to be done. However, after certain lectures and feedback from my tutor and spiritual mentor, it was prevailed and gave a good go at the data dictionaries, drawing up entity relationship diagrams and gaining a better understanding on normalisation. Still there is more to be learned in the normalisation department, but it is pretty normal to have weak points as a student and human being.

6. References

1. Agile development Tools, 7 August 2010, Agile development. Available at: http://agile-development-tools.com/ (Accessed 7 April 2013).Incremental

2. Agile development Tools, 7 August 2010, Scrum, Available at: http://agile-development-tools.com/agile-methodologies/scrum/ (Accessed 7 April 2013).

3. Biko technologies, 4 July 2009, Spiral Methodology. Available at: http://www.biko.com.au/about/how-we-work.asp (Accessed 7 April 2013).

4 Capper, G. Series of lectures from Teesside University blackboard System Design page Available at: https://eat.tees.ac.uk/webapps/portal/frameset.jsp?tab_tab_group_id=_2_1&url=%2Fwebapps%2Fblackboard%2Fexecute%2Flauncher%3Ftype%3DCourse%26id%3D_48485_1%26url%3D (Accessed 7 April 2013).

5. Hundhausen, R [2012] *Microsoft Professional Scrum Development with Microsoft Visual Studio 2012, edition 1,* Microsoft Press

6. MBA knowledge Base, 31 August 2010, the process of prototyping. Available at: http://www.mbaknol.com/management-information-systems/the-concept-of-prototyping-in-the-system-development-process/ (Accessed 7 April 2013).

7. Nawaz, M. Series of lectures from the Teesside University computing intranet staff page Available at: https://scm-intranet.tees.ac.uk/users/u0000706/ (Accessed 7 April 2013).

8. Philips, A. 4 Nov 2010, Software Development Methodology Available at: http://www.codeproject.com/Articles/124732/Software-Development-Methodologies (Accessed 7 April 2013).

9. Rising, J. 6 May 2009, Waterfall Software Development Process. Available at: http://www.managedmayhem.com/2009/05/06/sashimi-waterfall-software-development-process/ (Accessed 7 April 2013).

10. SpiraLabs, 7 August 2010, Rapid application development. Available at: http://www.spiralabs.com/?p=1 (Accessed 7 April 2013).

11. UML definition, Available at: http://www.motive.co.nz/glossary/uml.php, (Accessed 7 April 2013).

7. Appendix

Go to the link below ⬇

https://github.com/kanastasov/System-development-methods.git

For any of the following:

- ✓ **Low Level Data Flow Diagrams:**
- ✓ **Top Level Diagram:**
- ✓ **Context Diagram:**
- ✓ **UML Diagrams:**
- ✓ **Gant chart:**
- ✓ **The images for the methodologies:**
- ✓ **The normalization in .xlsx files:**
- ✓ **The report in .doc and .pdf files:**
- ✓ **The website with the master page, logo, including html, css and images:**
- ✓ **Algorithms for validation.**