

# SRS Software Requirement and Specification Home and Child Service

In the name of Allah, the most beneficent, and the most merciful.



[vumalik.blogspot.com](http://vumalik.blogspot.com)



All CS Free Services



LMS Handling

Quiz

GDB's

Assignment

Project



Contact Us

VU Malik

0317 6291353

To seek knowledge is the duty of every Muslim man and woman

Home and childcare services

Software requirements and specification

Version 1.0



**Group Id : mc123456789**

**Supervisor Name :Dr. Muhammad Salman Bashir**

### **Revision History**

<b>Date</b>	<b>version</b>	<b>Description</b>	<b>Author</b>
-------------	----------------	--------------------	---------------

16/12/2021	1.0	<p>Domestic and Child Care Administrations could be a web application that will permit its clients to enlist candidate for their homes. The candidate will be accessible for different administrations such as looking after children, servant, and cooking etc. In this web application, there will be 3 sorts of clients</p> <ol style="list-style-type: none"><li>1. Client</li><li>2. Candidate</li><li>3. Admin</li></ol> <p>The Candidate will be able to provide his/her availability in terms of days and timings and the minimum charges he/she will be charging the client with. Candidate will also be able to accept, reject or discuss on an offer made by the client.</p>	<b>mc123456789</b>
------------	-----	---	--------------------

## **Table of contents**

- ❖ **Scope**
- ❖ **Functional Requirements Non - Functional requirements**
- ❖ **Functional Requirement Diagram**
- ❖ **Usage Scenarios**
- ❖ **Adopted Methodology**
- ❖ **Work Plan**

## **Home and child care services**

### **Objective and Scope:**

Domestic and Child Care Administrations could be a web application that will permit its clients to enlist candidate for their homes. The candidate will be accessible for different administrations such as looking after children, servant, and cooking etc. In this web application, there will be 3 sorts of clients.

1. Client
2. Candidate
3. Admin

A Client ought to be able to enlist a Candidate through this web application. This web application will get the client's prerequisites within the frame of a survey and depending on the answers given by the client; the application will appear the closest coordinating Candidates to the client. Client will be able to send an offer to the Candidate of his/her choice. The Candidate will be able to supply his/her accessibility in terms of days and timings and the least charges he/she will be charging the client with. Candidate will too be able to acknowledge, dismiss or

examine on an offer made by the client. Taking after is the point-to-point data almost the client

## **Functional and Nonfunctional Requirements:**

### **2.1 What is Software & Engineering Requirements?**

#### **Software Requirements**

The software requirements are description of features and functionalities of the target system. Requirements convey the expectations of users from the software product. The requirements can be obvious or hidden, known or unknown, expected or unexpected from client's point of view.

The application ran on Apple devices. In this project our targeted audience is iPhone and iPad users at this stage.

#### **Requirement Engineering**

The process to gather the software requirements from client, analyze and document them is known as requirement engineering.

The goal of requirement engineering is to develop and maintain sophisticated and descriptive 'System Requirements Specification' document.

## **Types of Requirements**

Two type of requirements. These are following:

- Functional
- Non-Functional

### **Functional Requirements:**

Requirements, which are related to functional aspect of software fall into this category. They define functions and functionality within and from the software system.

#### **o Admin panel:**

- Admin should be able to add/update/remove a client or a Candidate manually.
- Admin should be able to accept/reject a client or a Candidate's sign up request. It means whenever a client or a Candidate signs up through the registration forms, they should not be added into the database until the admin approves.
- Admin should be able to create an offer on behalf of the client and should also be able to accept/reject an offer on behalf of a Candidate

#### **o Clients Panel:**

- A Client should be able to sign up using the Client Registration Form.
- The registration form should get the client's basic information and requirements.
- The basic information about the client should have the fields such as Name, Email, Phone, Address, Occupation etc.
- Client's requirements information will include if the client requires services of a babysitter, maid or a cook for the kitchen. It will also include the things like required experience, availability (in terms of days in a week and hours of the day), age, and gender of the Candidate.

- Client should have the interfaces such as hire a Candidate, Your Hiring History, and Suggestions (should show the Candidates who match the requirements), Update Your Account/Requirements, Logout etc.
- The suggestions area should show the closest matching Candidates to the client according to his/her requirements. The application will suggest the Candidates to the clients on the basis of their experience, age, gender, availability and charges (hourly/daily/weekly).
- Client should be able to send an offer to a Candidate in the list of suggestions. Once the offer is made by client, the status should show “Pending” until the Candidate accepts/rejects the offer. The status next to the offer should update accordingly after the offer is accepted/rejected.
- On the completion of this job, the client should be able to close the job on the site and give feedback on the Candidate’s performance.
- Once the job is closed, the information about this job should be moved to the Client’s Hiring History.
- Logout

#### **o Candidate Panel:**

- The Candidate should be able to sign up through a registration form.
- The registration form will get the basic information and the information about skills and availability information from the Candidate.
- Basic information will include the fields such as Name, Email, Phone, Address, Age, Gender etc.
- Skills and availability information will include Candidate’s interested job category (babysitter, maid or cook), availability (days of the week and hours in a day), how much they will charge (hourly/daily/weekly), experience etc.
- The Candidate’s site interface should have options such as Update Your Account (availability timings, charges etc), See the Offers, Previous Job History, and Logout etc.
- On the Offers page/section, the Candidates should be able to see the offers made by the clients for them, should be able to accept/reject an offer, and should be able to offer the client

his/her services on some different terms as compared to the client's offer.

- Once the offer is accepted/rejected, the status of the offer will be updated accordingly.
- On the completion of the job, the Candidate should be able to give feedback on the client according to client's cooperation, facilitation and other relevant parameters.
- Once the job is closed by client, it should move to the Candidate's previous job history.
- Logout

## ▪ **Non-Functional Requirements:**

**Nonfunctional Requirements (NFRs)** define system attributes such as security, reliability, performance, maintainability, scalability, and usability. They serve as constraints or restrictions on the design of the system across the different backlogs.

### ✓ **Performance:**

This is basically how quick your application functions. An exhibition prerequisite for the wayfinding application could be that it plots a course in under 20 seconds.

### ✓ **Responsiveness:**

This requirement ensures that your app is ready to respond to a user's input or an external event no matter what it's doing currently. For example, does your application permit the client to change to another capacity even while it's bustling ascertaining a course?

### ✓ **Scalability:**

Scalability is how well your app deals with increasing use, or size of data. For example, you may necessitate that the time your application takes to plot a course is direct with the distance between the two focuses.

### ✓ **Usability:**

This relates to how easily people can use your app. A measure of usability could be the time it takes for end users to become familiar with your app's functions, without training or help

### ✓ **Modifiability:**

This requirement governs how easily your app may be changed. For the wayfinding app, this requirement might state that the app can use more than one map without needing to be rebuilt.

### ✓ **Maintainability:**

This relates to the ease at which your app finds bugs and fixes them. For example, if a map is incorrect, can you fix the problem simply by downloading a correct map or do you have to purchase the next version of the app.

Cost is also often important. When you build an app, you must decide which nonfunctional requirements apply to your app. You also have to refine your NFRs so that you can test them. For example, for a performance NFR, you must be able to measure the speed of your application at the appropriate places (such as while the screen appears).

Certain nonfunctional requirements are especially important for mobile devices, such as performance and responsiveness.

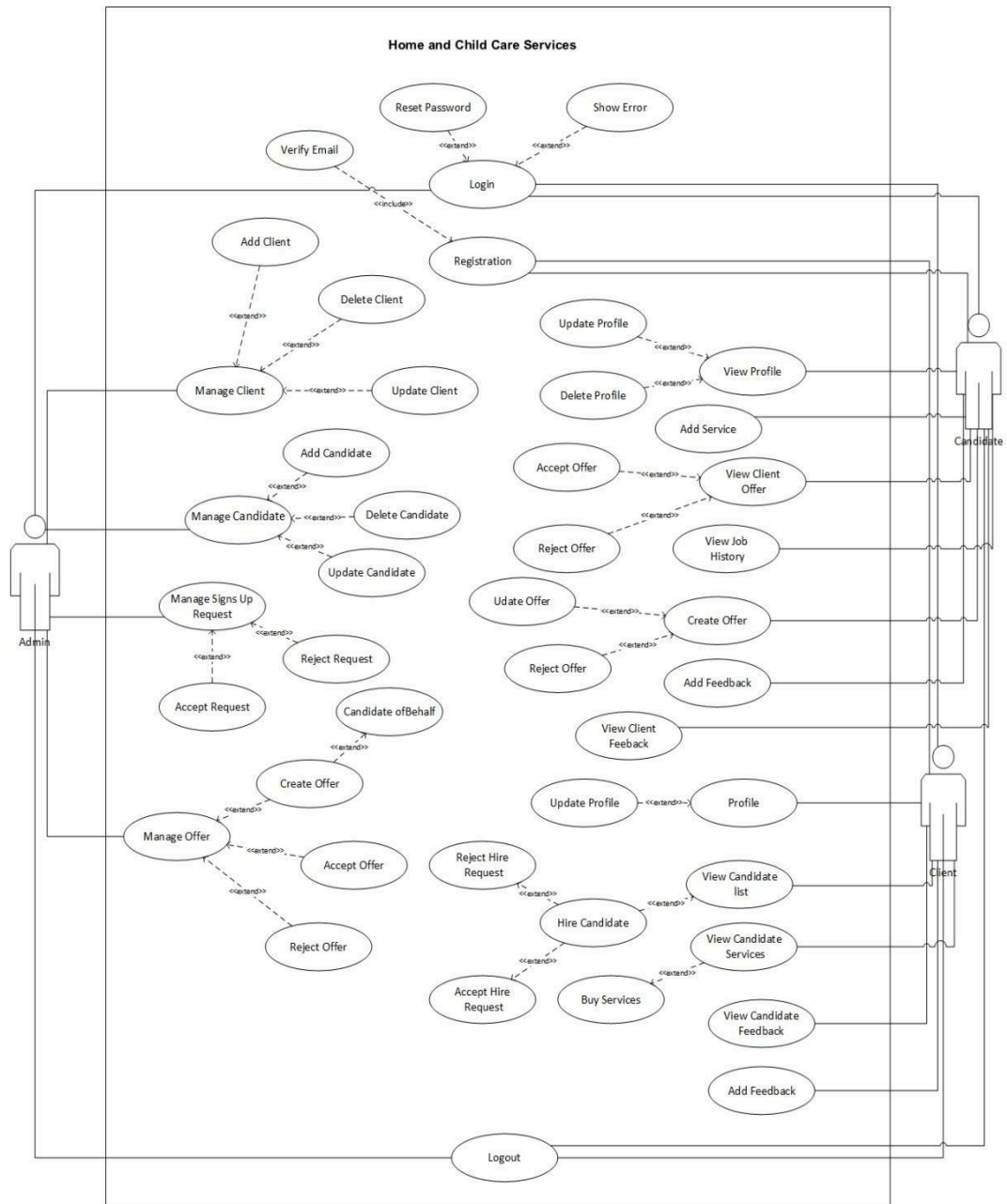
### ✓ **Reliability:**

This is the percentage of time that your app works correctly to deliver the desired results, despite potential failures in its environment. For the way finding app, it's the percentage of time that it correctly plots the route, given that the network may not be available or that bandwidth may be limited. You might require that your app work with a reliability of 99 percent, which means that in a given 24-hour day, it can fail to plot a route for no more than one percent of the day, or about 15 minutes.

### ✓ **Security:**

Your app saves all the previous routes it calculated and lets you reuse a saved route rather than recalculate it. If security isn't a requirement, you can store all app data in unencrypted files and on the SD card. If security is an issue, however, you can encrypt the files and store them in the app's private file area so that their content isn't easily accessible and so that they're deleted when the app is uninstalled

**Use Case Diagram(s):**



### Usage Scenarios:

<b>Functional Requirement title</b>	<b>Login</b>
<b>Functional Requirement Id</b>	<b>UC-1</b>
<b>Actor</b>	<b>Admin, client, Candidate</b>
<b>Actions</b>	All authorized users of this application will login.
<b>Description</b>	All users will login to system by using email and password for login to the systems. Users of this application will provide with email and password to enter in the required fields. Application will authenticate and allow only registered users for login
<b>Pre-Conditions</b>	All users must have application installed.
<b>Post Conditions</b>	All users must login after authentication
<b>Exceptions</b>	Application crash
<b>Author</b>	<b>MC123456789</b>

<b>Functional Requirement title</b>	<b>Manage client</b>
<b>Use case Id</b>	<b>UC-3</b>
<b>User</b>	<b>Admin</b>
<b>Actions</b>	Admin will login into the system application. All registered candidate will be shown in this panel to admin. Admin will select any candidate and update/ delete/block.
<b>Description</b>	<p>Admin will open the application. Admin will login to the system application by providing the credentials into the required fields. Admin will click on manage candidate button link. Admin will be shown with a panel where he will be shown with all registered candidate in the system. Admin will select any of the registered clients. He will update the information and submit the data. Candidate information will be updated and will be saved.</p> <p>For deleting the candidate information admin will select the user and he will click on the delete button. Candidate data will be deleted from the system.</p> <p>For blocking, admin select the candidate and blocked him.</p>
<b>Pre-Conditions</b>	User must be logged in.
<b>Post Conditions</b>	Admin should be able to manage the client.
<b>Author</b>	<b>MC123456789</b>
<b>Functional Requirement title</b>	<b>View user</b>
<b>Functional Requirement Id</b>	<b>UC-2</b>
<b>Actor</b>	<b>Admin</b>
<b>Actions</b>	Admin will login into the system application. He will open user panel. He will view the registered users
<b>Description</b>	Admin will open the application. Admin will login to the system application by providing the credentials into the required fields. Admin will click on view user button link. Admin will be shown with a panel where he will be shown with all added user in the

	system. Admin will view all the user details. He will view the details of the user by using this panel.
<b>Pre-Conditions</b>	Admin must be logged in.
<b>Post Conditions</b>	User must be registered by the admin
<b>Author</b>	<b>MC123456789</b>

<b>Functional Requirement title</b>	<b>Manage Signup Request</b>
<b>Functional Requirement Id</b>	<b>UC-4</b>
<b>Actor</b>	<b>Admin</b>
<b>Actions</b>	Admin will login into the system application. When the client or candidate sending a request for registration. Admin can be able approve or reject the request.
<b>Description</b>	Admin will open the application. Admin will login to the system application by providing the credentials into the required fields. Admin will click on registration request button link. Admin will be shown with a panel where he will be able to add new clients or candidates' details. Admin will register the client or candidate.
<b>Pre-Conditions</b>	Admin must be logged in.
<b>Post Conditions</b>	Candidate or client must be registered
<b>Author</b>	<b>MC123456789</b>

<b>Functional Requirement title</b>	<b>Manage candidate</b>
<b>Functional Requirement Id</b>	<b>UC-5</b>
<b>Actor</b>	<b>Admin</b>

<b>Actions</b>	Admin will login into the system application. All registered candidate will be shown in this panel to admin. Admin will select any candidate and update/ delete/block.
<b>Description</b>	Admin will open the application. Admin will login to the system application by providing the credentials into the required fields. Admin will click on manage candidate button link. Admin will be shown with a panel where he will be shown with all registered candidate in the system. Admin will select any of the registered clients. He will update the information and submit the data. Candidate information will be updated and will be saved.  For deleting the candidate information admin will select the user and he will click on the delete button. Candidate data will be deleted from the system.  For blocking, admin select the candidate and blocked him.
<b>Pre-Conditions</b>	Admin must be logged in.
<b>Post Conditions</b>	Client/Candidate data must be update and deleted respectively.
<b>Author</b>	<b>MC123456789</b>

<b>Functional Requirement title</b>	<b>Manage offer</b>
<b>Functional Requirement Id</b>	<b>UC-6</b>
<b>User</b>	<b>Admin</b>
<b>Actions</b>	User will login into the system application. He will click on create offer panel. Admin should be able to add, delete, reject accept and reject the offer on the behalf of candidate.
<b>Description</b>	User will open the application. He will login to the system application by providing the credentials into the required fields. Admin will click on manage offer button link. Admin should be able to manage an offer behalf of client/candidate.

<b>Pre-Conditions</b>	User must be logged in.
<b>Post Conditions</b>	Manage offer.
<b>Author</b>	<b>MC123456789</b>

<b>Functional Requirement title</b>	<b>View Candidate List</b>
<b>Functional Requirement Id</b>	<b>UC-7</b>
<b>Actor</b>	<b>Client</b>
<b>Actions</b>	User will login into the system application. He will click on view candidate list panel. All registered Candidate will be shown in this panel to user.
<b>Description</b>	User will open the application. User will login to the system application by providing the credentials into the required fields. User will click on view candidate list button link. User will be shown with a panel where he will be shown with all registered candidate in the system. Admin will select any of the registered client. He will update the information and submit the data. Client information will be updated and will be saved.  He will also be able to accept the hiring the request of candidate and also be able to reject the request of hiring candidate.
<b>Pre-Conditions</b>	User must be logged in.

<b>Post Conditions</b>	User data must be able to view the candidate list.
<b>Author</b>	<b>MC123456789</b>
<b>Description</b>	<b>Registration</b>
<b>Pre-Conditions</b>	<b>UC-9</b>
<b>Post Conditions</b>	<b>Client, Candidate</b>
<b>Author</b>	The users should be able to sign up through a registration form. The registration form should client's basic information and requirement. The basic information about the client/candidate should have the fields such as Name, Email, Phone, Address, Occupation etc.
<b>Description</b>	The users should be able to sign up through a registration form. The registration form should get the candidates/clients basic information and requirement. The basic information about the users should have the fields such as Name, Email, Phone, Address, Occupation etc. it sends the request to admin for signup. Then admin accept or reject the signup request.
<b>Pre-Conditions</b>	Admin must be logged in.
<b>Post Conditions</b>	User must be signup.
<b>Author</b>	<b>MC123456789</b>

<b>Functional Requirement title</b>	<b>Required services</b>
<b>Use case Id</b>	<b>UC-8</b>
<b>User</b>	<b>Client</b>
<b>Actions</b>	User will login into the system application. He will click on view services button. Clients mention his service in this panel like baby sitter, maid, and chef.
<b>Description</b>	User will open the application. User will login to the system application by providing the credentials into the required fields. User will click on required services button link. Admin will be shown with a panel where he will be shown all. It will also include the things like required experience, availability (in terms of days in a week and hours of the day), age, and gender of the candidates.

	And it also shows the need of the client like need of baby sitter, need of maid, need of chef.
<b>Pre-Conditions</b>	User must be logged in.
<b>Post Conditions</b>	View services must be shown
<b>Author</b>	<b>MC123456789</b>

<b>Functional Requirement title</b>	<b>Manage profile</b>
<b>Functional Requirement Id</b>	<b>UC-10</b>
<b>Actor</b>	<b>Client, candidate</b>
<b>Actions</b>	The user must be able to update his profile and account.
<b>Description</b>	The user must be able to manage his profile and account to his need. He can be update, delete, add, remove profile according to need.
<b>Pre-Conditions</b>	All Users must have application installed.
<b>Post Conditions</b>	All users must login after authentication.
<b>Exceptions</b>	Application crash
<b>Author</b>	<b>MC123456789</b>

<b>Functional Requirement title</b>	<b>View candidate services</b>
<b>Functional Requirement Id</b>	<b>UC-11</b>
<b>Actor</b>	<b>Client</b>
<b>Actions</b>	The user must be able to see the candidate services.
<b>Description</b>	The user must be able to view the candidate services.
<b>Pre-Conditions</b>	All Users must have application installed.
<b>Post Conditions</b>	All users must login after authentication.
<b>Exceptions</b>	Application crash

<b>Author</b>	<b>MC123456789</b>
---------------	--------------------

<b>Functional Requirement title</b>	<b>View candidate feedback</b>
<b>Functional Requirement Id</b>	<b>UC-12</b>
<b>Actor</b>	<b>Client, candidate</b>
<b>Actions</b>	The user must be able to view the candidate feedback.
<b>Description</b>	The Client must be able to view the feedback about candidate.
<b>Pre-Conditions</b>	All Users must have application installed.
<b>Post Conditions</b>	NA
<b>Exceptions</b>	NA
<b>Author</b>	<b>MC123456789</b>

<b>Functional Requirement title</b>	<b>View profile</b>
<b>Functional Requirement Id</b>	<b>UC-13</b>
<b>Actor</b>	<b>candidate</b>
<b>Actions</b>	The user must be able to view the profile.
<b>Description</b>	The user can be login. The user can be click on the view profile button. And he can be able to view the profile and update it. And also according to need, he can be delete it.
<b>Pre-Conditions</b>	All Users must have application installed.

<b>Post Conditions</b>	All users must login after authentication.
<b>Exceptions</b>	Application crash
<b>Author</b>	<b>MC123456789</b>

<b>Functional Requirement title</b>	<b>Add services</b>
<b>Functional Requirement Id</b>	<b>UC-13</b>
<b>Actor</b>	<b>candidate</b>
<b>Actions</b>	The user must be able to add services.
<b>Description</b>	The user can be login. The user can be click on the add services button.
<b>Pre-Conditions</b>	All Users must have application installed.
<b>Post Conditions</b>	All users must login after authentication.
<b>Exceptions</b>	Application crash
<b>Author</b>	<b>MC123456789</b>

<b>Functional Requirement title</b>	<b>Logout</b>
<b>Functional Requirement Id</b>	<b>UC-14</b>
<b>Actor</b>	<b>Client, candidate, Admin</b>
<b>Actions</b>	User can be logout any time.
<b>Description</b>	User can be logout any time.
<b>Pre-Conditions</b>	All Users have an account.
<b>Post Conditions</b>	All users must be login

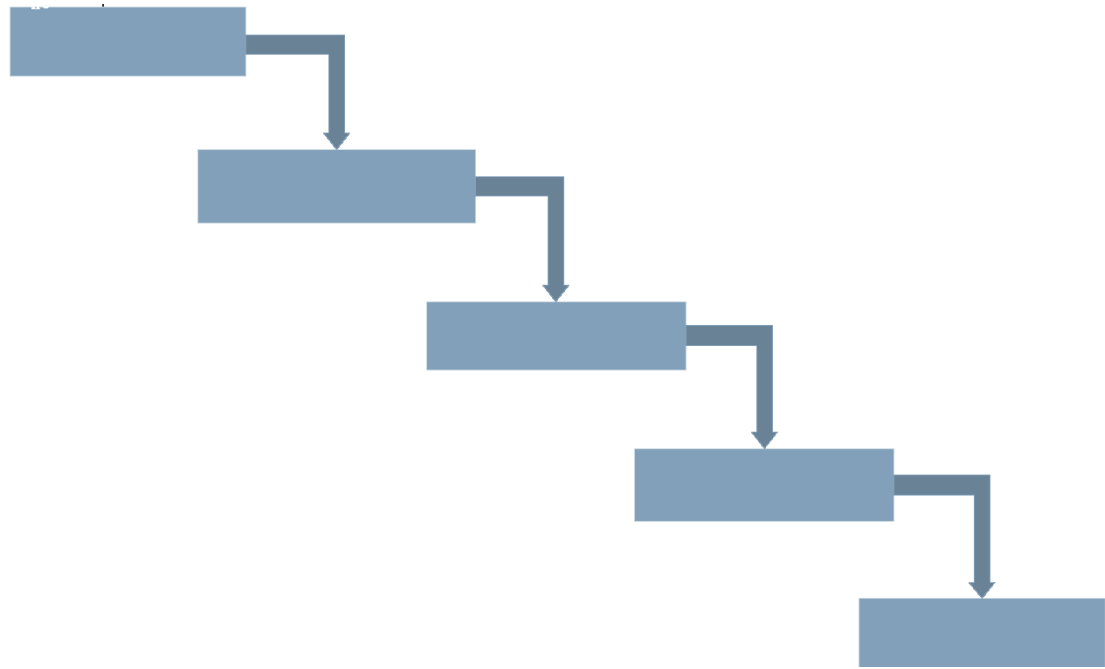
<b>Exceptions</b>	NA
<b>Author</b>	<b>MC123456789</b>

### **Adopted Methodology:**

There are various software development life cycles models defined which are followed during software development process.

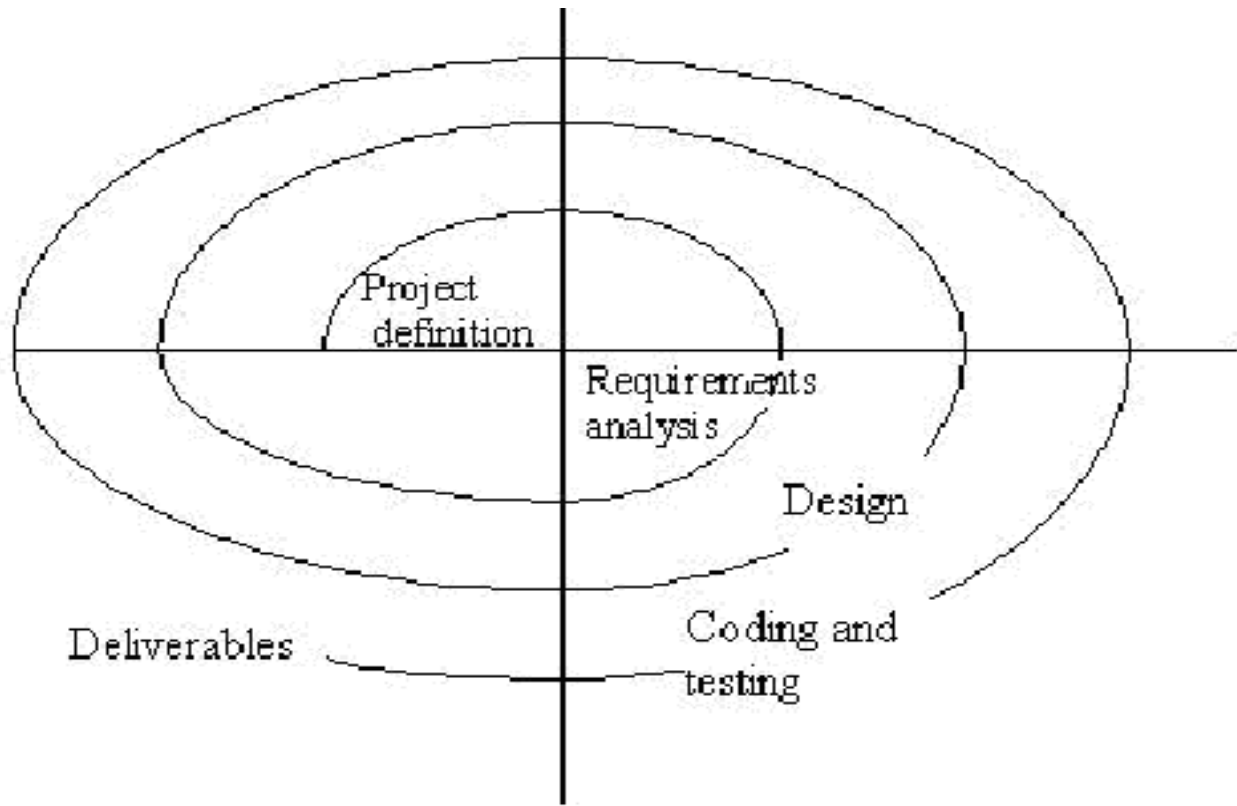
The waterfall approach is the early/earliest approach that was used for software development process. It is a linear sequential. Run steadily downwards through the phases of software implementation process. This means that any phase in the development process begins if and only if the previous phase is completed successfully. It is highly useful for developers because of following:

- Problems are more easily solved when they are defined first in any process.
- Large code will be more useful or tractable or better if it structured.
- Being a linear model, it is very simple to implement, the amount of resources required to implement this model are minimal/minimum.
- Creation of documents is done at every stage/phase of the software's development. It makes understanding of product design and procedures simple and easy.
- At every stage coding and testing is done to check the correct running of the code in development process.



### **Spiral Model:**

The Spiral Model is risk-driven process model. It is combination of both prototyping and waterfall model. It is an effort to combine the advantages of the both bottom up and top down concepts. It is often used for the large and complicated projects. There are four phases in it, which are: Planning, Evaluation, Risk Analysis and Engineering. All phases are iteratively followed one after other in order to eliminate all the problems



Spiral Process model

### **VU Process Model:**

VU process model is the combination of both Spiral and Waterfall process models proposed or developed by Virtual University of Pakistan in/for software development process. It explains the simplified description of software processes in cycle to

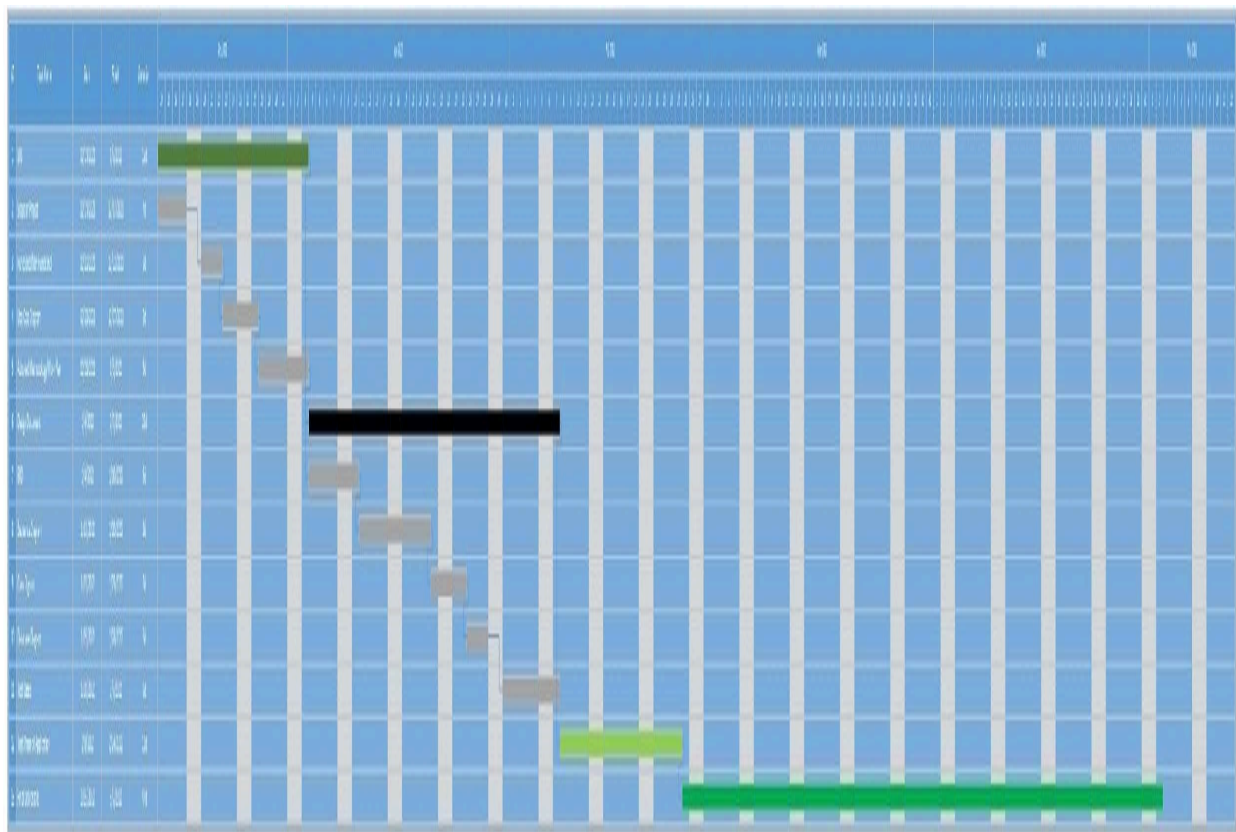
avoid/prevent maximum risk. It is called hybrid approach of system development because of combination of waterfall and spiral which maximize the quality of system and minimize the risks and disadvantages. It has four phases which are divided further in waterfall processes. These phases are in iteration to get the required specification for system or continue iteration until system cannot meet the system requirements.



### VU Model:

- It is best for Virtual University Projects.
- Testing is inherent in every phase.
- Clarity of projects vision and project Requirement is the essential features of the VU Model and I have clear cut project Requirements.

### Workplan:



**Tools:**

HTML, CSS

JavaScript, PHP

WAMP server,

SQL, Dream viewer, Notepad++