العلم سلاحی
LUMS

Software Requirements Specification Document (CS360)

Project Avocado

Group Number: 10

Maroof Azeem Saleemi
Muhammad Ali Shahzad
Muhammad Mustafa Asif
Muhammad Sabeeh Rehman
Omer Shakeel

Course: Software Engineering CS360

Instructor: Suleman Shahid

University: Lahore University of Management Sciences (LUMS

Version: 1.0
Date: (21/02/2020)
Number of hours spent on this document: 34

Contents

1 Introduction

1.1 Document Purpose

This document is for an app called Project Avocado. This is app provides a platform which serves the purpose of connecting clients and free lancers. The initial release will be 1.0. It will provide an ease for our client to manage his clients while also keeping a track of his employees and freelancers.

The main objective of this app is to give the clients and freelancers a single platform where they both can communicate with each other directly and avail services like content writing, web pages development and tutorials. This will be a mobile based app which will be available on the Play Store and the App Store after deployment.

1.2 Product Scope

Project Avocado is an app which provides a client with multiple options to select a freelancer for their appropriate job. There will be 3 main users of this app which will be the administrator, the client and the freelancer. The client will log in using their email and will specify the job they want done, in return the app will display a list of freelancers which can fulfill the job. The list will be sorted based on a rating system that will incorporate the reviews of the users, the duration of the freelancers and then experience. After the client has selected the job, he and the freelancer can contact through email and decide on the job. This will help our client manage his company as he can overlook everything through the app and doesn't have to get involved directly with the client and the freelancer.

1.3 Intended Audience and Document Overview

This document is intended for

1. The course instructor Suleiman Shahid.
2. The Teaching Assistant Assigned to us.
3. The client Abdullah Naseem.
4. The members of our group working on the project.

The rest of the document is divided into 4 sections.

The overall description will talk about the app and its perspective, functionality, user characteristics and the assumptions that we are making for this app

The Specific Requirements section will talk about functional and external interface requirements that our app will need. It also shows a use case diagram to give an idea about the app.

The non-functional Requirements sections will talk about the expected performance and the security features that our client will expect from us in this app.

Finally, there are four appendix which are as follows:

A. Top 10 user stories
B. Architectural spike
C. Group log
D. The contribution statements

2 Overall Description

2.1 Product Perspective

Project Avocado is mobile based application that allows the users to outsource the projects to free lancers in exchange for money. On Project Avocado, the users can easily find the range of top free lancers, from developers and development agencies to designers and marketers and much more. The platform would enable the free lancers to build portfolio from free lancing projects and market them to potential clients.

2.2 Product Functionality

2.2.1 Administrator

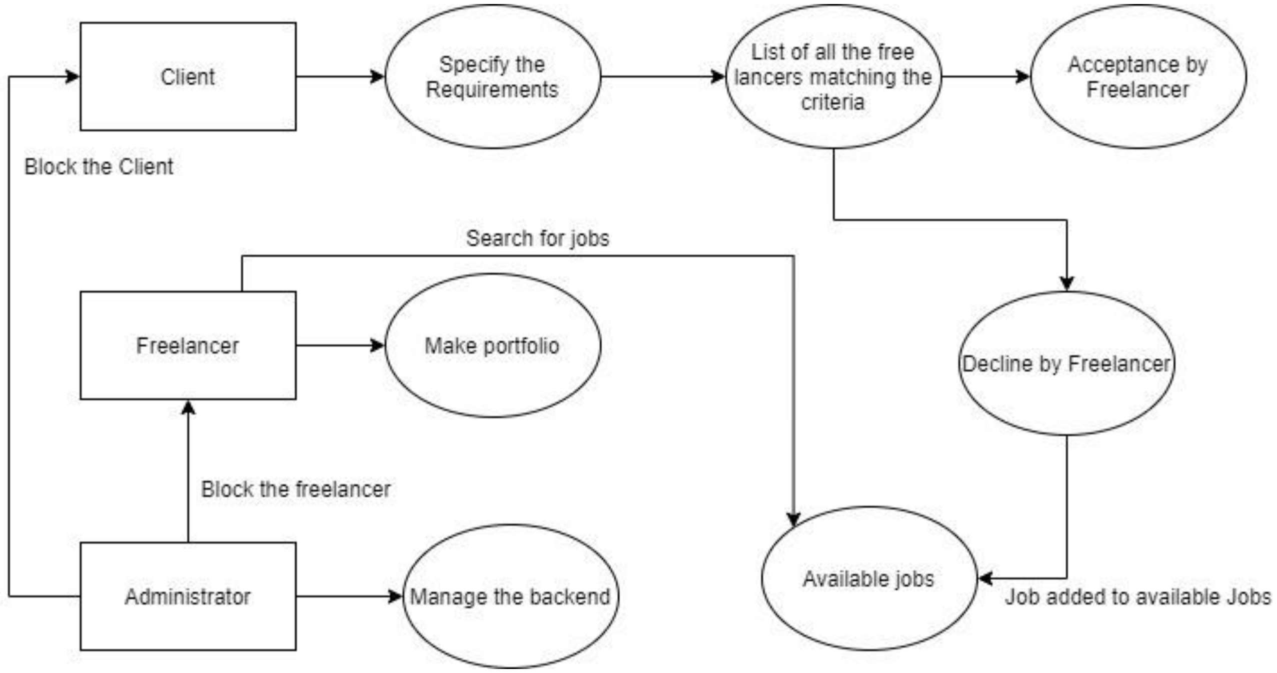
- Manage users and clients and block accounts if there is malicious activity
- Handle cases where client wishes to report a freelancer for malicious activity

2.2.2 Free Lancer

- Search for jobs: Free lancers would be able to search for a job related to their expertise and sort them based on various criteria such as compensation, due dates, job type, etc.
- Accept jobs: The app would allow the users to accept jobs which they find fit their expertise.
- Decline jobs: If the free-lancer feels that he cannot complete the work in the specified time period he would have the option to decline the offer.
- Personalise profile: They can add their expertise, profile picture and the previous work that they have done with other clients.

2.2.3 Client

- Post projects: Through the application the clients would post project in exchange for compensation.
- Specify project details: On posting a project the client will be prompted to enter the necessary details for the project such as the submission format and the software involved. He would also include the range that he is willing to pay for the project.
- Search for free lancers: The client will also be able to search for people and choose a potential free lancer who has the necessary expertise.
- View profiles: The client can see the people who have accepted the offers and then view their profiles. He/she can then screen from among those candidates based on rating, previous projects to assign the project to the best candidate.
- Offer the freelancer to complete the task: If the freelancer declines the offer, the client will have the option to go back to the free lancers list and search for another freelancer.



2.3 Users and Characteristics

There are three main types of users that interact with the system:

- The free lancers would use the application to search for the jobs, choose a job and navigate to it. This means that these users would mainly be focused on the "Jobs" page that lists all the free-lancing jobs. For the users to get all the relevant search details there are multiple search criteria that the users can specify and all results matches all of those.

- The clients could be large scale companies, small companies or even individuals looking to perform a specific task such as designing a logo for his Startup. They will list down description of the task, contact information and the skills needed.

- The administrators would also interact with the backend portal. They are managing the overall system so there is no incorrect information within it. The administrator can manage the information for each job as well as the options for both the free lancers and the clients.

2.4 Assumptions and Dependencies

- One assumption that we have about our application is that it would be used on cell phones with enough computational power and performance. The hardware resources needed by the application would be provided by the phones operating system to the application. However, if this is not the case then it is possible that the application does not work as it was intended.

- We also have an assumption that the users phone has a working camera so that he could post the picture along with the job posting.

- Furthermore, we are assuming that the phone would have a working internet access and the users will try to use the application only when it is connected to the internet.

- People will be willing to allocate time for the projects relevant to their field

- People will be educated enough to communicate their ideas efficiently in English.

3 Specific Requirements

3.1 Functional Requirements

Category: Login

RQ 1 – Client Functional Requirement 1	
Description	The system will allow the client to log into the application using their valid login
Input	User login details (username and password)
Processing	Authentication and saving the details
Output	‘User logged in’ message is displayed

Category: Signup

RQ 2 – Client Functional Requirement 2	
Description	The system will allow the client to create an account for the application
Input	• User login details (username and password) • First name, last name, phone number and email for signing up
Processing	Authentication and saving the details
Output	‘User Signed up message is displayed

Category: Add Job Request

RQ 3 – Client Functional Requirement 3	
Description	The system will allow the user to create a new task where they can choose from the freelancers visible for their task
Input	Name given to the job, nature of the job, requested finish date of the task and attaching document required for the job if applicable
Processing	Saves the task details and creates a new task in the basket
Output	Creates a new tab for submission for a file or to indicate if task is completed

Category: Browse Available Freelancers

RQ 4 – Client Functional Requirement 4	
Description	The system will allow the user to choose from the available freelancers suitable for their job
Input	User to select a freelancer from the visible options
Processing	Generates a request for the freelancer chosen
Output	Freelancer is notified about the job offer and a ‘Request Sent’ message is displayed

Category: Accept/Decline Request

RQ 5 – Client Functional Requirement 5	
Description	The system will allow the user to accept or decline the freelancer request
Input	Selecting the accept or reject button with optional box of reason for rejecting
Processing	Adds the job in the freelancer's basket if accepted or if rejected send message to the freelancer
Output	Freelancer is notified about the request reply from the client and a 'Request Accepted' or a 'Request Declined' message is displayed to the Client

Category: View All Jobs Posted

RQ 6 – Client Functional Requirement 6	
Description	The system will allow the user to view the list of pending jobs or the history of the tasks completed for the users themselves
Input	Selecting the button from the side drawer through which the history is displayed
Processing	Requests the jobs corresponding to the user viewing it from the database
Output	Shows the job in order of newest (including pending) to oldest

Category: View Job Detail

RQ 7 – Client Functional Requirement 7	
Description	The system will allow the user to view a pending job they posted themselves
Input	Selecting the job from the list of all available jobs
Processing	Requests the job corresponding to the user viewing it from the database
Output	Shows the job detail

Category: Rate Freelancer

RQ 8 – Client Functional Requirement 8	
Description	Rate the freelancer and view current rating
Input	Tap on the icon to rate and then submit the rating
Processing	Depending on the number of icons selected out of 5, the rating is sent to the database
Output	Rating and rank of the freelancer is updated

Category: Communication

RQ 9 – Client Functional Requirement 9	
Description	Communicating with the freelancer application support through external application
Input	Selecting the help option and then choosing report or selecting email the user
Processing	Requests email of the user/admin to facilitate communication
Output	Prompts a link to Outlook/Gmail with an email addressed to the requested user

Category: View Profile

RQ 10 – Client Functional Requirement 10	
Description	The system will allow the user to view the freelancer profile
Input	The user will be given option to select a freelancer from the list and view their profile
Processing	User profile info will be fetched from the database
Output	Shows the freelancer profile

Category: Login

RQ 11 – FreeLancer Functional Requirement 1	
Description	The system will allow the freelancer to log into the application using their valid login
Input	User login details (username and password)
Processing	Authentication and saving the details
Output	‘User logged in’ message is displayed

Category: Signup

RQ 12 – FreeLancer Functional Requirement 2	
Description	The system will allow the freelancer to create an account for the application
Input	• User login details (username and password) • First name, last name, phone number and email along with additional information - Area of expertise and an attachment of their resume
Processing	Authentication and saving the details
Output	‘User Signed up message is displayed

Category: Browse Available Job

RQ 13 – FreeLancer Functional Requirement 3	
Description	The system will allow the user to choose from the available jobs suitable for their area of expertise
Input	The freelancer to select a job from the visible options
Processing	Generates a request for the job chosen
Output	Client is notified about the offer and a ‘Request Sent’ message is displayed to the Freelancer

Category: Accept/Decline Request

RQ 14 – FreeLancer Functional Requirement 4	
Description	The system will allow the user to accept or decline the client request
Input	Selecting the accept or reject button with optional box of reason for rejecting
Processing	Adds the job in the freelancer’s basket if accepted or if rejected send message to the client
Output	Client is notified about the request reply from the client and a ‘Request Accepted’ or a ‘Request Declined’ message is displayed to the FreeLancer

Category: View All Jobs Taken

RQ 15 – FreeLancer Functional Requirement 5	
Description	The system will allow the user to view the list of pending jobs or the history of the tasks completed by the users themselves
Input	Selecting the button from the side drawer through which the history is displayed
Processing	Requests the jobs corresponding to the user viewing it from the database
Output	Shows the jobs in order of newest (including pending) to oldest

Category: View Job Detail

RQ 16 – FreeLancer Functional Requirement 6	
Description	The system will allow the user to view a pending job they are working on
Input	Selecting the job from the list of all available jobs and selecting the 'Job Completed' option when the job is completed
Processing	Requests the job corresponding to the user viewing it from the database and if job is completed, updates the database and notifies the client
Output	Shows the job detail and Prompts a 'Task Completed' message to the client when job is completed

Category: Communication

RQ 17 – FreeLancer Functional Requirement 7	
Description	Communicating with the client application support through external application
Input	Selecting the help option and then choosing report or selecting email the user
Processing	Requests email of the user/admin to facilitate communication
Output	Prompts a link to Outlook/Gmail with an email addressed to the requested user

Category: Edit Portfolio

RQ 18 – FreeLancer Functional Requirement 8	
Description	The system will allow the user to edit and update their profile
Input	The user will be given option to edit their bio and their portfolio information accordingly
Processing	User profile info will be updated in the database accordingly
Output	Shows a 'Profile Updated' message and their profile will display changes.

3.2 External Interface Requirements

3.2.1 User Interfaces

A simple GUI is required for this software where user can touch to interfere with the software.

Fig1.1

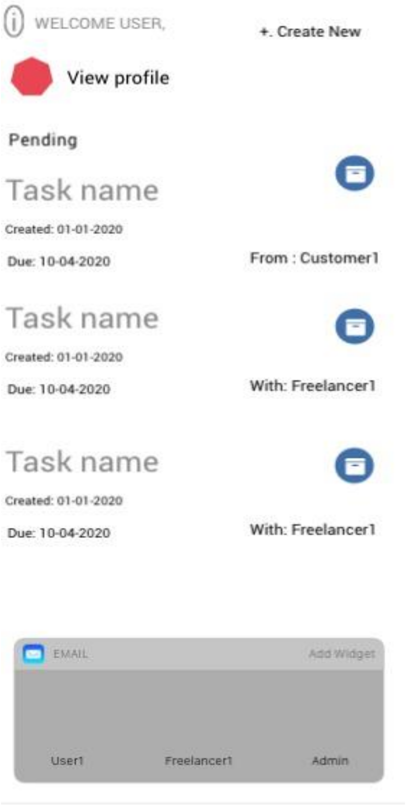


Fig1.2

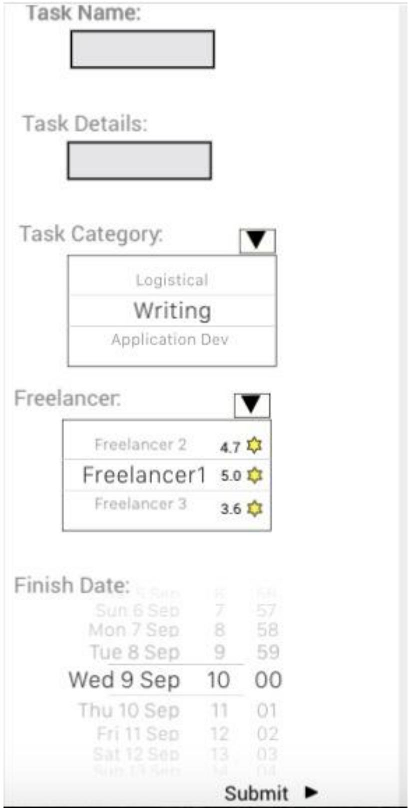


Fig1.3

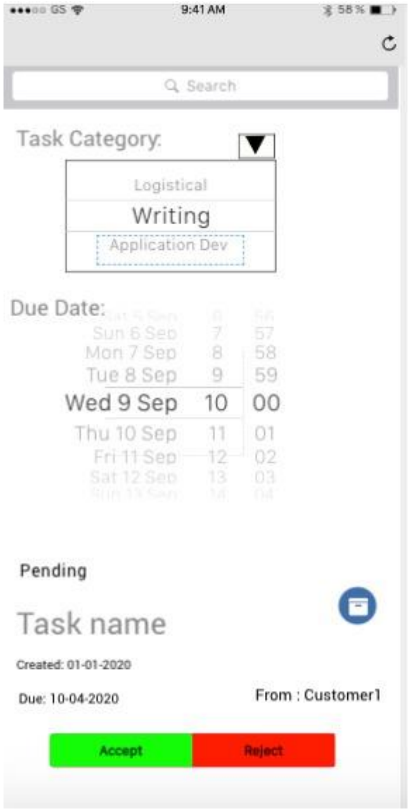


Fig 1.1. Client is shown the home screen where they can view pending tasks, shown an option for help, view profile, create new task and email the required freelancer or software admin.

Fig 1.2. Client is shown the create new task screen where they can give the task name, task details, choose task category and due date through a drop-down box. They can select the freelancer while viewing their ratings.

Fig 1.3. Freelancer is shown the search screen where they can search for tasks available and accept/reject tasks requested by clients to them.

3.2.2 Hardware Interfaces

- Android and IOS devices are to be supported by this software.
- Android devices with API 16 (android version 4.1 or higher) will be required for the software.
- IOS devices will be required to have a version of 9.0 or above.
- Android SDK will be used in order to communicate the software to android.
- React native bridge would be sufficient for communication with IOS.

3.2.3 Software Interfaces

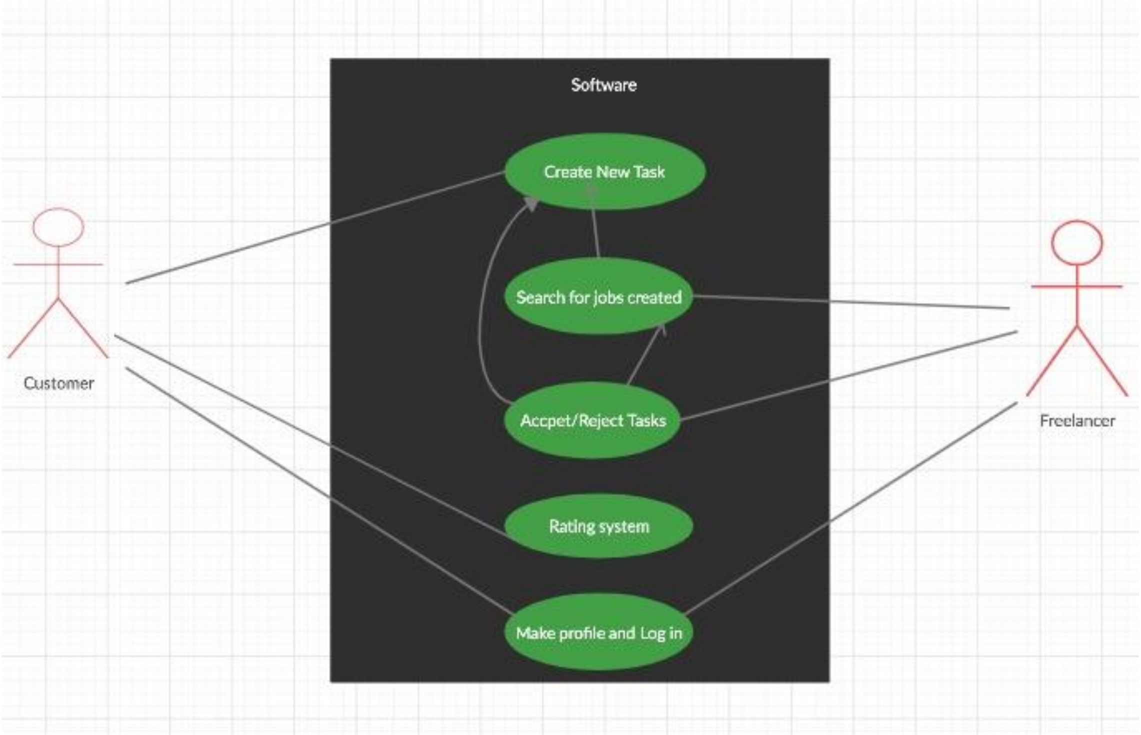
SQL based database will be used and MySQL Connector (python library) will be required to communicate with the database. In order to communicate the front software with the database, Flask will be used. React native and its libraries will be used to communicate the software with the operating system. Android studio will be the interface used with the operating system.

3.3 Use Case View

3.3.1 Use Case Table

Primary Actor	Associated Use cases	
Client	1.Make profile 2.Create a new Task 3.View pending tasks/history of tasks 4.View/ update profile 5.Rate Freelancer 6.Be able to login and sign-up. 7.View details of freelancer 8.Accept/reject freelancer request	
FreeLancer	1.Accept and reject a task 2.Search for tasks 3.Send request to be assigned task 4.Make profile 5.Login and signup 6.View history of tasks and tasks at hand	

3.3.2 Use Case Diagram



3.3.3 Use Case Description

Use Case 1: Create a new Task

Client will create a new task by providing the details and name. They will decide the category which will help narrow down the freelancers who will be able to undertake the task. The client will then be able to choose the freelancer he/she wants in particular and a request will be sent to that freelancer. The freelancer can accept or reject this task depending on the details etc.

Use Case 2: Search for Tasks

All freelancers can search for tasks apart from the tasks that they have been requested. The returned search result will be a list of tasks based on the category the freelancer and the earliest due date the freelancer chose while searching. The freelancer who has searched for these tasks will be able to send request to the client indicating that he/she is willing to undertake the task.

Use Case 3: Rate Freelancer

After the task is completed, the client will be able to rate the freelancer. The client will choose rating through a star based rating method. The number of icons the user taps on will be the rating of the freelancer out of 5.

Use Case 4: View previous tasks

The client and freelancer both will be able to view the tasks they have completed or have to complete or even have a request for. The pending applications will be visible on the main page whereas the history will be available in the drawer on the main page. The corresponding ratings, date and cost of the task will also be visible to the user.

Use Case 5: Make profile

The user will be required to make a profile where they have to enter their details and choose between being the client or freelancer. The user will have to enter the first name, last name, phone number, email and other information. This profile will be visible to the other users who need to request work.

4 Other Non-functional Requirements

4.1 Performance Requirements

1. **Response Time**
   The application should load and process queries in Industry Standard time, i.e. the user generated queries should not take more than 7 seconds to process

2. **Workload**
   The application must be able to support multiple users operating the app. It should also be able to support multiple queries generated by several clients.

3. **Scalable**
   The application should be able to scale to the growing user database. It should be able to cater to new clients as well as new freelancers.

4. **System**
   The application will run both on Android and iOS devices.

5. **Database Update**
   The database should be updated whenever a client or freelancer want to browse the available jobs or freelancers

4.2 Safety and Security Requirements

4.2.1 Safety

1. The possibility of harm, loss, or damage from the use of the application are highly improbable being that the product is intended for use in what is assumed to be controlled environment. Also, the planned utilization of the item doesn't present many predictable situations where a client would experience damage of any sort in request to satisfy the maximum utility of the product.

2. In the scenario where the freelancer has monetarily exploited the client while failing to comply to the demands of the client, the client has the option either report the freelancer or to give the freelancer a bad review which would impact the freelancer’s future client job prospects.

3. Similarly, if a client unfairly rates the freelancing user, they will have to submit a proper reasoning for it which will be examined by the admins.
4. Only the clients of a freelancer have the option to rate them, this would ensure a fair rating system.

5. A new freelancing user must enter further information such as qualification and speciality, this can be done through submitting a resume along with their application. After that would only be accepted if approved by the admin themselves.

4.2.2 Security

Since almost all the application features require a network connection, a certain degree of user security is expected to prevent unsolicited access to the user information. This will be achieved through the following steps:

1. The application will use 2-step authentication system to register a new user which will require the user to enter some basic personal information – name, email, date of birth, area of expertise (applicable to a new freelancer), etc. along with a password of minimum 8 characters to set up a new account. The user will be sent an authentication request link through the email which would complete the authentication and the registering process of the new user.

2. An existing user will be given access based on a secured authorization in which they will be required to enter their registered email and password.

3. The server will be hosted using secure and credible web server to maintain a respectful level of security as all the user database will be stored at the server.

4. To avoid any unwanted access to the database separate database schemas will be used as permissions on multiple schemas and schema-contained objects can be managed with greater precision than in a single-schema oriented database.

5. The application will not disclose any personal information of the client.

4.3 Software Quality Attributes

4.3.1 Correctness

The development team will make sure that the specifications that determine how users can interact with the application and how the application should behave when it is used correctly are in sync with each other. The problem will be defined completely and only then will the algorithm and the problem logic will be developed. Proved models will be used, where applicable, to maximize the effectiveness. Developers will pay attention to the clarity and simplicity of the program while verifying each part of a program as soon as it is developed.

4.3.2 Reliability

Most functionality will require a stable network connectivity. System components that require authentication will be processed through the internet and network connectivity will function if and only if the server is constantly running at the backend. This will be achieved using a reliable web-based server.

4.3.3 Availability

Rather than making the application exclusive to any specific platform, the application will be available on Android as well as iOS which will increase the user-reach of the application. The server will always be running to extend the availability window for the functionalities of the app.

4.3.4 Maintainability

The development team will follow best practices for clean code and programming modularity to make the application as viable as could be expected under the circumstances.

4.3.5 Adaptability

For the application itself to be adaptable the software architecture used to make the application will be adaptable, it will be able to figure out which platform the application is running on, whether iOS or Android and render the compositions suitable for that platform.

4.3.6 Portability

Users will be able to access this application anywhere using their smart devices given they have a reliable internet connection available.

4.3.7 Reusability

The application will be reusable for both the clients and the freelancers as they have the option to login into their account and make use of the functionalities Irregardless of whether they have used the app before or not.

4.3.8 Extensibility

The application will be highly extensible in terms of adding new features such as adding a payment method, or a client freelancer chatting feature in the future if needed

4.3.9 Compatibility

The application shall be compatible with Android version 5.0 Lollipop or higher and iOS version 10 or higher. As for hardware compatibility, the application doesn't require high computational power and will suffice with a smart device which can run simple apps like that of Foodpanda.

4.3.10 Usability

- A simple to use User Interface will be used for the application with an appropriate green colour scheme as requested by the client.
- The font used in the application must be in uniform.
- Easy navigation will be provided along with labeled objects which will be understandable for the users.
- The homepage will be used to navigate to all the different features the application provides. The user has the option to go back to the homepage from any screen.

Appendix A – Top 10 User Stories

1. As a client, I want to be able to add descriptions, attach files and comprehensively describe the task I want done while posting about it, so that the client I choose has a good understanding of the work I expect before deciding to accept my offer.

2. As a client, I want to be able to view a list of freelancers with expertise related to my task so I can choose one to assign my work to.

3. As a client, I want the freelancers to be ranked according to their average ratings as well as the level of experience they have so that I can gauge their competence before assigning my work to them.

4. As a client, I want to be able to view how much a particular client charges so that I can make a decision based on my budget.

5. As a client, I want to be able to rate the freelancer I worked with once the task is done so that it can help future clients to make informed decisions.

6. As a client, I want my task to be placed on a list for pending tasks, in case my preferred client chooses to reject my offer so that the task can still be taken up by other freelancers with relevant expertise.

7. As a client, I want to have the option to accept or decline an offer by a freelancer for my job that was placed on the list for pending jobs so that I do not have to work with a client whose rating or charges are not ideal.

8. As a freelancer, I want to be able to make myself available for work in any field of my choice by specifying my interest and attaching my resume so that any relevant clients can be directed to me.

9. As a freelancer, I want to have the option to accept or decline an offer of work by a client so that I can choose the jobs that best suit my needs

10. As a freelancer, I want to be able to browse through jobs relevant to my expertise that have not been taken up yet so that I can place an offer for a job in case it suits me.

Appendix B – Architectural Spike (One Story)

User Story 3, where the client states, “As a client, I want the freelancers to be ranked according to their average ratings as well the level of experience they have so that I can gauge their competence before assigning my work to them,” falls under the “Architectural Spike” as we think there could be some problems in deciding how much weightage to give to each factor while ranking the freelancers. Furthermore, we will have to deal with outliers in terms of ratings as we would not want one bad rating to affect the position of a client in a list. Hence, we plan to figure a way around this problem before we begin implementation and possibly take a look at existing systems that have implemented similar functionality.

Appendix C - Group Log

- We conducted four weekly meetings with our assigned Teaching Assistant so we could share our progress and plan out the document with her. Each meeting was roughly between 15 and 20 minutes long.

- Furthermore, we met thrice with our client, for around half an hour on each occasion, once to establish clarity over the requirements of the project, once to go over the document a final time with him and the last time to ask a few questions regarding the application

- Lastly, we met among ourselves four times so we could divide the tasks, share our work and ensure it linked up. These meetings were longer (around a couple hours).

Appendix D – Contribution Statement

Name	Contributions in this phase	Approx. Number of hours	Remarks
Maroof Azeem Saleemi	Mostly worked on compiling the functional requirements and describing use cases, creating a use case diagram for assistance and so on.	9	-
Muhammad Mustafa Asif	Articulated a holistic description of the project, describing the scope and functionality of the project, the assumptions behind it, the different types of users and their potential roles and interactions with the application.	6	-
Muhammad Ali Shahzad	Focused on putting together the non-functional requirements of the project, which include the safety and security requirements, performance requirements and the quality expectations.	8	-
Muhammad Sabeeh Rehman	Compiled the introduction where he gave an overview of the entire project and the document and touched upon the different details that were later elaborated in the document. Assigned work to the team members and linked it up.	6	-
Omer Shakeel	Collected user stories, listed ten of the most significant ones in this document, categorized one as falling under “Architectural Spike”. Furthermore, kept track of meeting minutes, and individual contributions behind this document.	5	-