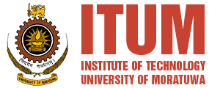
****

**Institute of Technology University of Moratuwa**

**ADVAISA**

**Software Requirements Specification**

**Version 1.0**

*Student Details:*

G.H.D. Amarasena 21IT0456

K.W.B.B.J. Kevitiyagala 21IT0486

K.A.D.T. Lalantha 21IT0493

K.K. Mithishi 21IT0499

K.A.D.B.N. Nanayakkara 21IT0501

*Supervised by:*

Ms. M S Madhubhashini

Division of Information Technology

Institute of Technology University of Moratuwa

April 2024

# Table of Contents

[Table of Contents 1](#_Toc164841608)

[1 Introduction 3](#_Toc164841609)

[1.1 Purpose 3](#_Toc164841610)

[1.2 Product Scope 3](#_Toc164841611)

[1.2.1 Aim and Objectives 3](#_Toc164841612)

[1.2.2 Project Boundary 3](#_Toc164841613)

[2 Overall Description 4](#_Toc164841614)

[2.1 Product Perspective 4](#_Toc164841615)

[2.2 User Classes and Characteristics 4](#_Toc164841616)

[2.3 Operating Environment 4](#_Toc164841617)

[2.4 Design and Implementation Constraints 4](#_Toc164841618)

[2.5 Assumptions and Dependencies 4](#_Toc164841619)

[3 External Interface Requirements 5](#_Toc164841620)

[3.1 User Interfaces 5](#_Toc164841621)

[3.2 Hardware Interfaces 5](#_Toc164841622)

[3.3 Software Interfaces 5](#_Toc164841623)

[3.4 Communications Interfaces 5](#_Toc164841624)

[4 System Designs 6](#_Toc164841625)

[4.1 Use case Diagram 6](#_Toc164841626)

[4.1.1 Use case Description 6](#_Toc164841627)

[5 System Features 14](#_Toc164841628)

[5.1 System Feature 1 14](#_Toc164841629)

[5.1.1 Description and Priority 14](#_Toc164841630)

[5.1.2 Stimulus/Response Sequences 15](#_Toc164841631)

[5.1.3 Functional Requirements 15](#_Toc164841632)

[5.2 System Feature 2 15](#_Toc164841633)

[6 Other Nonfunctional Requirements 15](#_Toc164841634)

[6.1 Performance Requirements 15](#_Toc164841635)

[6.2 Safety Requirements 16](#_Toc164841636)

[6.3 Security Requirements 16](#_Toc164841637)

[6.4 Software Quality Attributes 16](#_Toc164841638)

[7 References 16](#_Toc164841639)

[Appendix 17](#_Toc164841640)

[Figure 2: Example for the High -Level Diagram 18](#_Toc164841671)

[Figure 3: Example for the user and their characteristics 18](#_Toc164841672)

[Figure 4: Example for the use case narratives 20](#_Toc164841673)

[Figure 5: Example for the system feature 21](#_Toc164841674)

**No table of figures entries found.**

# Introduction

## 1.1 Purpose

This document focuses on describing the software requirements of the project, a Mobile application for experts and user connecting platform named ‘ADVISA’. This project would be carried out to meet the software development requirements of the IT 2308 module conducted by the Division of Information Technology Institute of Technology University of Moratuwa. The document focuses on the software design and all the functional, non- functional, performance, security, and safety requirements of the project.

## 1.2 Product Scope

### 1.2.1 Aim and Objectives

**Aim**

The aim of our project is developing a mobile application that enhances the consultation experience by providing users with a virtual platform to connect with expert professionals.

**Objective**

* Protect user privacy by anonymous posting.

Ensure that users can post their questions or concerns anonymously, safeguarding their identity and personal information.

* User-friendly interface.

Design an interface that is intuitive and easy to navigate, allowing users to access the application's features without confusion or difficulty.

* Quick access to qualified professionals.

Provide users with a platform where they can quickly connect with verified and experienced professionals in their desired field of consultation.

* Promoting transparency.

Ensure that the platform operates with transparency, providing clear information about the professionals, their qualifications, and the consultation process.

* Scheduling appointments.

Enable users to easily schedule appointments with professionals based on their availability and convenience.

* Seamless communication.

Facilitate seamless communication between users and professionals, allowing for real-time messaging or video calls to ensure effective consultation.

### Project Boundary

A mobile consultation application's boundaries specify its parameters and outline what is and is not included in the program. The limits aid in creating a clear understanding of the function and features of the program.

The parameters of a mobile application for consultation are as follows:

* Functionality for Consultations: The main goal of the consultation mobile application is to offer features for client and consultant consultations. This offers functions including logging consultations, communication channels, managing client profiles, and appointment scheduling. Through effective communication and collaboration between consultants and clients, the application seeks to improve and expedite the consultation process.
* User Management: The application includes user management functionality to handle the registration, authentication, and authorization of consultants and clients. It allows users to create accounts, log in securely, and manage their profiles. User management features enable personalized experiences, secure access, and appropriate role-based permissions within the application.
* Communication and Messaging: The consultation mobile application includes communication and messaging capabilities to facilitate real-time or asynchronous interactions between consultants and clients. It may incorporate features like chat, video conferencing, or secure messaging channels. These communication channels enable consultants and clients to communicate effectively and securely within the application.
* Appointment and Scheduling: The application provides features for scheduling appointments between consultants and clients. It allows clients to view consultant availability, book appointments, and receive confirmations. Consultants can manage their schedules, view upcoming appointments, and send reminders to clients. The application may include notifications to keep both parties informed about appointment status and updates.
* Profile Management: The application enables users to manage their profiles, including personal information and preferences. Consultants can showcase their expertise, qualifications, and availability. Clients can provide their contact details, preferences, and relevant information. Profile management features ensure that consultants and clients can maintain accurate and up-to-date information within the application.
* Data Security and Privacy: The consultation mobile application places a strong emphasis on data security and privacy. It incorporates measures to protect user data, including encryption, secure authentication, and compliance with relevant privacy regulations. The application ensures that user information and consultation data are handled securely and confidentially.

It is noteworthy that the bounds of the mobile application for consultation do not encompass any other features or functionalities that are not directly associated with the consultation process. The consultation mobile application would not cover, for instance, unrelated social networking elements, unconnected e-commerce functions, or other unrelated services. Stakeholders can better grasp the focus, aim, and features of the consultation mobile application by establishing clear boundaries around it. This clarity guarantees that the application's declared scope is met by both user expectations and development efforts.

# 2 Overall Description

## 2.1 Product Perspective

The consultation application simplifies collaboration between consultants and clients by integrating various architectural elements. The User Interface (UI) provides a user-friendly interface, while the Consultation Management component manages user authentication and access controls. The Communication and Messaging component facilitates real-time communication between clients and consultants, while the Data Management component ensures data security and integrity. The Integration and APIs component enables connectivity with external systems, such as payment gateways. The Analytics and Reporting component collects and analyzes data related to consultations, user behaviors, and application usage, providing valuable insights for improvement. To visually represent the architectural components, various diagrams can be used, including high-level system architecture diagrams, detailed component diagrams, and sequence diagrams. These diagrams help stakeholders understand the application's structure, interactions, and interactions, ensuring clarity and effective communication among project stakeholders.

## 2.2 User Classes and Characteristics

User Classes

The following user classes have been identified for our system.

* System Administrator
* Experts
* User

User Characteristics

* System Administrator

The System Administrator is responsible for the overall management and maintenance of the consultation mobile application. They have privileged access to the system and are responsible for tasks such as installing and configuring the application, managing user accounts, monitoring system performance, and ensuring the security of the application. The System Administrator is also responsible for handling any technical issues or troubleshooting that may arise and ensuring the smooth operation of the application.

* Experts

Professionals are qualified experts offering consultation services in their respective fields. They create profiles showcasing their expertise, qualifications, and availability for consultations. Their role is to provide consultation services, answer user queries, schedule appointments, and communicate effectively with clients. Professionals can create and manage their profiles, set their availability, respond to user queries, and conduct consultations through the platform.

* User

Users are individuals seeking consultation services for various needs such as legal advice, mental health support, beauty advice, nutritionists support, or career guidance. They create an account on the platform to access services and can interact with professionals through messaging or video calls. Their role is to seek assistance, ask questions, schedule appointments, and engage in consultations with professionals. Users can access the platform's features such as posting queries, viewing professional profiles, scheduling appointments, and messaging with professionals.

## 2.3 Operating Environment

* Hardware Platform:

The System Administrator is responsible for the overall management and maintenance of the consultation mobile application. They have privileged access to the system and are responsible for tasks such as installing and configuring the application, managing user accounts, monitoring system performance, and ensuring the security of the application. The System Administrator is also responsible for handling any technical issues or troubleshooting that may arise and ensuring the smooth operation of the application.

* Operating System and Versions:

Experts are someone who have specific knowledge and experience in a given field or area. They are essential in the consultation mobile application since they help users who are looking for information or assistance by offering consultation services. Experts are usually experts or extremely knowledgeable people who can respond to users' questions or problems with insightful advice or answers. They might possess particular training, credentials, or work experience in their specialized fields. Through the program, experts interact with users, offer consultations, and sometimes even add to the system's resources or knowledge base articles.

* Coexistence with Other Applications:

On the user's mobile device, the consultation mobile application should live in harmony with other installed software apps. It shouldn't interfere with other widely used programs, including web browsers, email clients, messaging apps, or social media sites, or cause performance problems when used in tandem with them. To improve the user experience, compatibility and integration with platform-specific features and services, such sharing capabilities or push alerts, may also be taken into consideration.

In general, mobile applications for consultation should be developed and designed to guarantee optimal performance, compatibility, and a smooth user experience within the designated hardware and software environment. To preserve the functionality and performance of the application over time, it is crucial to execute routine upgrades and compatibility tests with new operating system versions and relevant software components.

Technology Stack

* Front-end :

React Native

JavaScript

* Back-end:

ExpressJs

NodeJs

* Database:

MySQL

* Version Control:

git

## 2.4 Design and Implementation Constraints

* Corporate or Regulatory Policies:

Developers must adhere to corporate policies or regulatory guidelines set by the organization or industry. These policies may dictate certain security standards, data privacy regulations, or compliance requirements that need to be followed during the development process. Developers must ensure that the application meets these policies and guidelines.

* Hardware Limitations:

The mobile application for consultation must take the target devices' hardware restrictions into account. These restrictions may be related to computing power limitations, memory needs (such as RAM limitations), or timing requirements (such as response time). To ensure a seamless user experience, developers must optimize the application to operate effectively within these hardware constraints.

* Interfaces to other applications:

The application may need to integrate with other applications or services, such as authentication services, messaging platforms, or payment gateways. Developers need to ensure that these integrations are seamless and secure.

* Specific technologies, tools, and databases:

The customer or organization may have specific requirements for the technologies, tools, and databases to be used in the application development. Developers need to adhere to these requirements while ensuring that the chosen technologies are suitable for the application's needs.

* Parallel operations:

The application may need to perform parallel operations, such as handling multiple user requests simultaneously or processing background tasks. Developers need to implement these operations carefully to avoid performance issues and ensure data integrity.

* Language requirements:

Regarding the application's linguistic needs, developers might be limited. This may entail localizing the application for particular target markets or geographic areas or supporting a number of languages. Developers must incorporate suitable language support and localization frameworks since language constraints might affect user interface, content translation, and localization efforts.

* Communications protocols:

The application may need to communicate with servers or other devices using specific communications protocols, such as HTTP, HTTPS, or WebSocket. Developers need to ensure that the application follows these protocols to establish and maintain communication.

* Security considerations:

The application must implement robust security measures to protect user data and prevent unauthorized access. This includes encryption of sensitive data, secure authentication mechanisms, and adherence to security best practices.

* Design conventions or programming standards:

The customer's organization may have specific design conventions or programming standards that developers need to follow. This ensures consistency and maintainability of the code base, especially if the organization will be responsible for maintaining the delivered software.

## 2.5 Assumptions and Dependencies

Assumptions

* Third-Party Components: It is anticipated that particular third-party components—like frameworks, libraries, or APIs—will be used in the application's development. It is assumed that these parts will be accessible, compatible, and perform as intended. Any changes in functionality, unavailability, or compatibility problems with these components may have an effect on the application's requirements and implementation.
* Development Environment: The assumption is made that the development environment, including development tools, IDEs, compilers, and specific software versions, will remain stable and accessible throughout the project. Changes in the development environment might require adaptations to the development process and affect the application's requirements.
* Operating Environment: It is assumed that the target operating systems and devices, such as iOS and Android, will remain consistent with the versions specified in the SRS. Changes in these operating systems and devices might introduce new requirements or impact the compatibility of the application.
* User Behavior and Usage Patterns: The requirements stated in the SRS are based on assumed user behaviors and usage patterns. These assumptions include factors such as user preferences, expectations, and typical usage scenarios. If these assumptions are incorrect or change, it might affect the design, functionality, and user experience of the application.

Dependencies

* External APIs: For a number of functions, including data retrieval, payment processing, and authentication, the project may be dependent on external APIs. For the program to be implemented successfully, these APIs' reliability, compatibility, and availability are essential.
* Database Systems: For the storage and retrieval of data, the project may be dependent on particular database systems or database management tools. Modifications or restrictions in these database systems may affect the application's operation and demand for data.

Reusable Software Components: The project might make use of modules or software components designed to be transferred from one project to another. These components' functioning, compatibility, and availability are essential to completing the project on schedule and within the required budget.

# 

# 3 External Interface Requirements

## 3.1 User Interfaces

Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.

Eg: mockups of the user interfaces: login page, etc. (Draw at-least 3 basic user interfaces)

## 3.2 Hardware Interfaces

Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.

Eg: RFID reader

## 3.3 Software Interfaces

Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.

Eg: Google Map API, Payment gateway API, Shopping cart libraries, Server operating system

## 3.4 Communications Interfaces

Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.

Eg: HTTP, SMTP, FTP (Name the protocol and reasons of using it)

# 4 System Designs

This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.

## 4.1 Use case Diagram

A diagram of a company structure

Description automatically generated

### 4.1.1 Use case Description

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | (prefix your Use Case IDs with UC, to distinguish them from other models) | |
| **Use Case** | Login | |
| **Priority** | High | |
| **Primary System Actor** | Admin, User , Expert | |
| **Other participant Actors** | - | |
| **Description** | Users need to login to the application to use the main functionalities of the system | |
| **Pre-conditions** | User must register to the system | |
| **Trigger** |  | |
| **Flow of Events** | Actor Action | System Response |
| Step 2: Enter Username  Step 3: Enter password.  Step 4: Press login button | Step 1: Display login interface  Step 5: Display message “Login successfully” |
| **Post-conditions** | Display home page | |
| **Alternative Flows** | Step 2 and Step 3: The system detects multiple failed logins attempts from the same IP address or account. Step 5: The user is prompted to use the "Forgot Password" feature. | |
| **Assumptions** | Messages are displayed by the system. | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | (prefix your Use Case IDs with UC, to distinguish them from other models) | |
| **Use Case** | Verify Expert | |
| **Priority** | High | |
| **Primary System Actor** | Admin | |
| **Other participant Actors** | - | |
| **Description** | Admin needs to identify the true and verified expert. Admin should verify their authorization before register the profile. | |
| **Pre-conditions** | Admin must login to the system | |
| **Trigger** |  | |
| **Flow of Events** | Actor Action | System Response |
| Step 2: Verify the authorization.  Step 3: Accept the request. | Step 1: View the request.  Step 4: Send an email that mentions that their request is approved. |
| **Post-conditions** | Send an email with relevant information to the applicant. | |
| **Alternative Flows** | Step 2: If the documents are invalid or fake, the request will be rejected. Step 4: Send an email that mentions that their request is rejected. | |
| **Assumptions** | Emails are sent by the system. | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | (prefix your Use Case IDs with UC, to distinguish them from other models) | |
| **Use Case** | Blog Posting | |
| **Priority** | High | |
| **Primary System Actor** | Expert | |
| **Other participant Actors** | - | |
| **Description** | Expert can Post the blog to their own profile | |
| **Pre-conditions** | Expert must login to the system | |
| **Trigger** |  | |
| **Flow of Events** | Actor Action | System Response |
| Step 1: Go to the Profile  Step 2: Upload the blog | Step 3: Display blog |
| **Post-conditions** | Blog post successfully | |
| **Alternative Flows** | Step 2: If the step 2 Success user and other expert can view the blog | |
| **Assumptions** | Elogs are display | |

|  |  |  |
| --- | --- | --- |
| **Use Case** | Read Blog posts | |
| **Priority** | medium | |
| **Primary System Actor** | Expert, User | |
| **Other participant Actors** | - | |
| **Description** | Users and Experts can read the blog post | |
| **Pre-conditions** |  | |
| **Trigger** |  | |
| **Flow of Events** | Actor Action | System Response |
| Step 1: Go to the Blog section.  Step 3: View and read the post | Step 2: Display the blog post. |
| **Post-conditions** | View the Blogs | |
| **Alternative Flows** | Step 2: If the step 2 Success user and other expert can view and read the blog | |
| **Assumptions** |  | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | (prefix your Use Case IDs with UC, to distinguish them from other models) | |
| **Use Case** | Post Problem | |
| **Priority** | High | |
| **Primary System Actor** | User | |
| **Other participant Actors** | - | |
| **Description** | Users can Post the their problem Anonymous or normal | |
| **Pre-conditions** | Users must log in to the System | |
| **Trigger** |  | |
| **Flow of Events** | Actor Action | System Response |
| Step 1: Go to the post section.  Step 3: Users select a method (Anonymous or normal).  Step 4: Type and Post the Problem | Step 2: Display the post section.  Step 5: View the that post |
| **Post-conditions** |  | |
| **Alternative Flows** | Step 2: If the step 3 and step 4 Success user and other Expert can view this problem post give the solution | |
| **Assumptions** | Problem posts are Display in System. Successfully Messages are displayed by the system. | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | (prefix your Use Case IDs with UC, to distinguish them from other models) | |
| **Use Case** | Chat | |
| **Priority** | High | |
| **Primary System Actor** | User , Expert | |
| **Other participant Actors** | - | |
| **Description** | Users and experts can chat about their problems and solutions through this app. | |
| **Pre-conditions** | Users and expert  must log in to the System | |
| **Trigger** |  | |
| **Flow of Events** | Actor Action | System Response |
| Step 1: Go to the chat  section  Step 3: Both can chat via the chat secton | Step 2: Display the chat section |
| **Post-conditions** |  | |
| **Alternative Flows** |  | |
| **Assumptions** | Chat display on the system | |

Include use case scenarios or narratives which describe how users will perform functions in the system. Here each use case is represented as a sequence of step identified with pre-conditions and post-conditions. (*Refer the Appendix C template)*

Eg: Refer the Appendix D

# 5 System Features

## 5.1 System Feature 1

Don’t really say “System Feature 1.” State the feature name in just a few words.

Eg: Refer the Appendix E

### 5.1.1 Description and Priority

Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).

### 5.1.2 Stimulus/Response Sequences

List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.

### 5.1.3 Functional Requirements

Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1:

REQ-2:

## 5.2 System Feature 2

# 

# 6 Other Nonfunctional Requirements

## 6.1 Performance Requirements

If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.

## 6.2 Safety Requirements

Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.

## 6.3 Security Requirements

Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.

## 6.4 Software Quality Attributes

Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.

# 7 References

List down all the sources (including web sites, journals, articles, books) you've used in your project, so readers can easily find what you've cited.

Follow IEEE guidelines when including references in your research paper

<https://ieee-dataport.org/sites/default/files/analysis/27/IEEE%20Citation%20Guidelines.pdf>

# Appendix

Append any other figures, tables or documents related to your document. Properly mention each and every item that you are appending.

Each appendix should deal with a separate topic

* Each appendix must be referred to by name (Appendix A, Appendix B, Appendix C, etc.)
* If there is only one appendix, it is just called Appendix
* Each appendix must also have a title
* Begin each appendix on a separate page with page number

**Appendix A**

**Diagram

Description automatically generated**

Figure 2: Example for the High -Level Diagram

**Appendix B**

**Text

Description automatically generated**

Figure 3: Example for the user and their characteristics

**Appendix C**

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | (prefix your Use Case IDs with UC, to distinguish them from other models) | |
| **Use Case** |  | |
| **Priority** | (High, Medium or Low) | |
| **Primary System Actor** |  | |
| **Other participant Actors** |  | |
| **Description** | (brief) | |
| **Pre-conditions** |  | |
| **Trigger** |  | |
| **Flow of Events** | Actor Action | System Response |
| 4. If xxx 5. Else 6. etc. |  |
| **Post-conditions** |  | |
| **Alternative Flows** | (briefly describe alternative flows here for base Use Cases; extend this into complete flow of events for Elaborated Use Cases, either here or in the next template) | |
| **Assumptions** |  | |

**Appendix D**

Table

Description automatically generated

Figure 4: Example for the use case narratives

**Appendix E**

**Text, letter

Description automatically generated**

Figure 5: Example for the system feature