
Software Requirements Specification to Academic Advisor Website

Version 1.0 approved

Prepared by:

Arwa Almrzoqi

441212001

Hissah Almousa

441211996

Munirah Aldaosari

441211997

**Qassim University
Software Engineering and Knowledge Engineering CSC606.
Instructor:
Dr. Faisal Alhwikem.**

1/3/2023

Table of Contents

Table of Contents	ii
List of Figures.....	ii
List of Tables	ii
1. Introduction	1
1.1 Purpose	1
1.2 Scope	1
1.3 References	2
1.4 Structure	2
2. System Overview	2
2.1 Product Perspective	2
2.2 Product Features	3
2.3 User Roles and Characteristics	3
2.4 Operating Environment	4
2.5 Assumptions and Dependencies	4
3. External Interface Requirements	4
3.1 User Interfaces	4
3.2 Hardware Interfaces	5
3.3 Software Interfaces	5
3.4 Communications Interfaces	5
4. System Requirements	5
4.1 System Functional Requirements	6
4.2 System Non-Functional Requirements	6

List of Figures

Figure (1): Software architecture	3
---	---

List of Tables

Table (1): Functional Advisor Requirements	6
Table (2): Functional Student Requirements	6

1. Introduction

The introduction of the Software Requirements Specification (SRS) provides an overview of the whole SRS document including purpose, scope, references, structure, system overview, interfaces, functional requirements, and non-functional requirements. The aim of this document is to gather and analyze the Academic Advisor Website System in-depth and give a full insight into the system.

1.1 Purpose

The purpose of this document is to present the Academic Advisor Website in a detailed description. The document explains the ideas that have been collected and analyzed to define the system of the proposed software and its constraints and goals. This document defines the target user and its interface as well as the hardware and software interfaces. In addition, it clarifies how the users see the software and what functionalities are supported. In addition, what the software provides and how it operates and reacts accordingly under some predefined constraints.

1.2 Scope

The software is the Academic Advisor Website for students and advisors at Qassim University and is based entirely on them. This system is designed to provide services for both student and advisor as well as the contribution of AI for significant invention and benefits.

More specifically, this software copes with the difficulties of arranging an appropriate time to meet with the academic advisor for any reason. The student is able to view the advisor's availability and make an appointment as needed. On the other hand, the advisor views the appointments and manage them as well by accepting or rescheduling. Additionally, the advisor updates his schedule constantly in order for both to be aware and avoid conflicts. This software helps the student to get in touch with the advisor without actually annoying him by sending messages back and forth for that purpose.

The system also contains the AI which is connected to the student's database involving the student program, plan, schedule, and all related data. That contributes for a great benefit in order to minimize the physical real meetings and less time consuming.

1.3 References

- [1] Alhwikem, Dr. F. (n.d.). *Software Requirements* [Slide show]. https://quedusa-my.sharepoint.com/:b:/r/personal/faisal_alhwikem_qu_edu_sa/Documents/Semesters/442/CS606/Slides/CS606_Lecture6.pdf
- [2] Alison Weber. (2016, September 11). *Software Engineering: Chapter 3 SRS Explained* [Video]. YouTube. <https://www.youtube.com/watch?v=FvxO8SLoua8>
- [3] IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications*. IEEE Computer Society, 1998.

1.4 Structure

The remaining sections of this document provide all the required information that clarifies the full idea of the software. The next section, System Overview, gives an overall description of the system. It describes the product perspective and features, the characteristics and roles of the users, and what are some assumptions and dependencies of the system. The third section discusses the external interface requirements involving the user, hardware, software, and communication interfaces. The final section is for the requirements of the whole system. This section gives a detailed description of the functional requirements and includes the non-functional requirements that deal with the efficiency, security, dependability, and usability of the system.

2. System Overview

This section of the SRS describes the general factors that affect the product and its requirements. It discusses the factors in detail in the next subsections.

2.1 Product Perspective

The Academic Advisor website is a sub-component that cannot work on its own but requires access to end-users MyQU pages. However, the MyQU page represents the central part of all the available sub-components that make up the entire services provided through the MyQU page.

The Academic Advisor sub-component does not directly have an interface that connects the two end users. Instead, each student has a chance to contact an AI chatbot if that connection does not minister to the student, he will book an appointment and go to communicate and consult his/her

academic advisor directly during his/her office hours. Figure 1 presents an overall view of the software architecture. The website, on its own, has two interfaces: one to the academic advisors sub-component and another one to the student sub-component. With this, we want to present that, the two different interfaces the academic advisor and student provided in the figure are externals and have an API. As well, there is no interface to the database as it is internal.

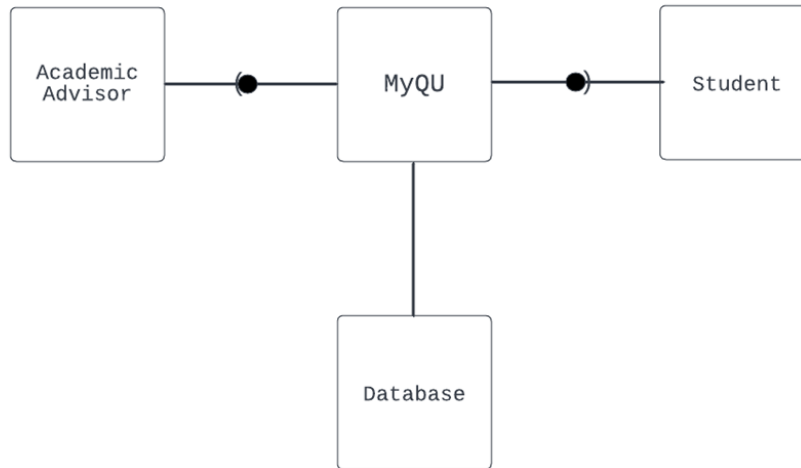


Figure (1): Software architecture.

2.2 Product Features

For an Academic Advisor Website, the features might include easy access to the Academic Advisor Website and its availability to all university students, by accessing the student's page through MyQU. Furthermore, it facilitates communications between advisors and students, in which academic advisors will prepare and update their available office hours easily, also, the academic advisors will have an updated status about the new students. In addition, the most beneficial feature is the provision of time and effort for students and advisors through the presence of an assistant AI Chabot. Whereas, if student inquiries are not adequately answered by the Chabot, the system will provide an opportunity for students to book an appointment with their academic advisor and communicate with them directly.

2.3 User Roles and Characteristics

Users of the software system include students, academic advisors, and software developers. All users should be familiar with how to use web technology. As well, some users must be present members of Qassim University, either academic advisors or students. In addition, users are should be knowledgeable of how to use and benefit from the provided services on the MyQU website. However, users should have a good understanding of how to use the tasks and activities in which they may be interested effectively.

2.4 Operating Environment

The server-side components of the software system must operate within a Qassim University operating system environment.

The client-side components of the software system must operate within common web browser environments using SHA-256 with RSA Encryption. The set of browsers that supported are Apple Safari, Google Chrome, Microsoft Internet Explorer, and Mozilla Firefox, etc.

2.5 Assumptions and Dependencies

One of the assumptions and dependencies on which the proposed software is based may be that it depends on that all users firstly should access their personal page on the Qassim University MyQU website to reach and use the proposed software. Users cannot access the proposed software unless they access and login to their MyQU personal pages.

3. External Interface Requirements

This section of the SRS describes the interface requirements for the system. Requirements for user, hardware, software, and communication interfaces defined.

The only link to an external system is the link to the Qassim University (MyQU) Database to allow the users to log in and the system get the required data from the database.

3.1 User Interfaces

The interface of the software is easy to understand and compatible to any browser by which user can access to the system. The components of the user interface are explained in details below:

- Screen Layout:

The main screen allows the user to log in either as a student or an advisor. Then the user transferred to a page to fill out the username and password considering that the student page has its own services and functionality as well as the advisor's page.

- Functions:

The advisor's page has the options of services as follows: manage my schedule, my students, and my appointments.

The student's page has the options of services as follows: view my advisor schedule, and ask for appointment. Additionally, the student is able to interact with AI chatbot. This invention provides the student with his plan, schedule, future courses, and any information that is accessible in the database. That will be helpful to reduce the physical meetings as possible in case of having any concerns regarding the student's program.

3.2 Hardware Interfaces

This software is developed to work on Android, IOS mobile devices as well as the computers supported by Windows, Linux, and Mac. No other hardware is required. Server-class computers must be used to run all server-side components. Workstation-class and personal-class computers are required for the operation of all client-side components.

3.3 Software Interfaces

The software interfaces designed to run and supports several operating systems such as Apple macOS, Microsoft Windows, Google's Android OS, Linux Operating System, and Apple iOS. In addition, the fundamental languages used for both front-end and back-end development will be define and customize.

- Server side:

An Apache web server will accept all requests from the client and forward them accordingly. A database will hosted centrally using MyQU database.

- Client side:

A website that is capable of running a modern web browser that supports JavaScript, CSS and HTML.

3.4 Communications Interfaces

The communication architecture must follow the client-server model. Communication between the client and server should utilize a REST-compliant web service and must be served over HTTP Secure (HTTPS). The client-server communication must be stateless. A uniform interface must separate the client roles from the server roles. The server may send data from its database as HTML, XML, or JSON.

4. System Requirements

The following section presents the complete set of functional and non-functional requirements identified for the subject academic advising website. Functional requirements are listed first, according to their relationship to students and advisor. The non-functional requirements that pertain to availability, security, usability and reliability.

4.1 System Functional Requirements

This subsection presents the identified functional requirements for the academic advising website. Initially, advisor requirements that pertain to the system are given.

4.1.1 Advisor

Table (1): Functional Advisor requirements.

Requirement	Description
A01	Advisors shall be able to log in the system and log out.
A02	Advisors shall be able to manage their time table (free, busy, Teaching).
A03	Advisors shall be able to manage students they supervise.
A04	Advisors shall be able to View a list of students they supervise.
A05	Advisors shall be able to Review a list of appointments they have with students.

4.1.2 Student

Table (2): Functional student requirements.

Requirement	Description
S01	Students shall be able to log in the system and log out.
S02	Students shall be able to Review the schedule of supervisors who supervise them.
S03	Students shall be able to Book appointments with supervisors commensurate with their times.
S04	Students shall be able to interact with the Chabot.

4.2 System Non-Functional Requirements

This subsection presents the identified non-functional requirements for the subject academic advising website. The subcategories of non-functional requirements given are availability, security, usability and reliability.

4.2.1 Availability:

The software should be available anytime and anywhere.

4.2.2 Security:

The software must protect its users from hacking and secure their personal and academic information in addition to encrypting their data and preventing external access to it from others.

4.2.3 Usability:

The software must be designed in a way that allows it to be used without training and in record time.

4.2.4 Reliability:

The software must carry out the tasks for which it was designed and identified in the analysis phase effectively and efficiently.