SWE 216

Software Requirements Specification (SRS)

The document in this file is an annotated outline for specifying software requirements, adapted from the IEEE Guide to Software Requirements Specifications (Std 830-1993).

Tailor this to your needs, removing explanatory comments as you go along. Where you decide to omit a section, you might keep the header, but insert a comment saying why you omit the data.

Software Requirements Specification

for

KFUPM Tutoring

Version 3.6 approved

Prepared by Team 6

KFUPM

16 APR 2022

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
SAUD ALFARIS	26 FEB 2022	FIRST RELEASE	1.0
SAUD ALFARIS	14 FEB 2022	- REQUIREMENTS UPDATE, TO FIT USECASES.	1.1
KHALED ALBUGAMI	15 FEB 2022	- ADDED USECASE DESCRIPTIONS	1.5
SAUD ALFARIS	16 FEB 2022	- ADDED ACTIVITY DIAGRAMS AND NON-FUNCTIONAL REQUIREMENTS	2.0
SAUD ALFARIS	12 MAY 2022	- ADDED USER INTERFACE REQUIREMENTS AND SIMULATIONS	3.1
SULTAN ALMOAMMAR	11 MAY 2022	- ADDED HARDWARE INTERFACE REQUIREMENTS	3.2
KHALED ALBUGAMI	11 MAY 2022	- ADDED SOFTWARE INTERFACE REQUIREMENTS	3.3
ABDULLAH ALHABIB	12 MAY 2022	- ADDED COMMUNICATION INTERFACE REQUIREMENTS	3.4
TURKI ALZAHRANI	12 MAY 2022	- ADDED APPENDIX A	3.5
MUTAZ ALSHAHRANI	12 MAY 2022	- ADDED APPENDIX B	3.6

1. Introduction

1.1 Purpose

The purpose of KFUPM Tutoring is to help students help themselves, or to assist or guide them to the point at which they become an independent learner, and thus no longer need a tutor. Our software makes that easier to both learners and tutors in KFUPM. Enabling them to preform mundane tasks automatically.

1.2 Document Conventions

The SRS document uses few different font sizes for clear distinction. For example, the main headings are numbered with whole numbers like 1. Introduction 2. Overall Description. The subheadings are numbered with decimals like 1.1 Purpose, 1.2 Document conventions

1.3 Intended Audience and Reading Suggestions

The document is intended to be read by course instructor, users and documentation writers. The document is organized into 5 parts. 1. Introduction, 2. Overall Description, 3. System Features, 4. External Interface Requirements, 5. Quality Attribute Requirements. All the parts are independent however reading the document sequentially helps the reader understand the KFUPM Tutoring system better.

1.4 Project Scope

Our team has decided to give a new perspective to the tutoring experience in KFUPM. That should help to reduce the cost of time and money.

- Goals
    - Improve the student learning experience,
    - Introduce a new field of learning,
    - Provide income for students who are willing to be tutors.
- Deliverables
    - More flexibility in terms of time and place for tutoring to be held for both the student and the tutors.
    - Enabling academically suffering students.
    - Raising the quality of current tutoring supply.
    - Enhancing competition among tutors.

2. Overall Description

2.1 Product Perspective

The KFUPM Tutoring mobile application is new product focused on creating a new tutoring experience in KFUPM, increasing the level of quality in the local tutoring market.

2.2 Product Features

- Creation of accounts for tutors and learners
- Users can change their profile details (picture, expertise, etc.)
- Tutors can specify the tutoring subject/course, time, rate, type of tutoring (one to one, or one to many), etc.
- Learners can search for available tutors/subjects/courses.
- Learners can book a tutorial and pay in advance an agreed upon percentage.
- Learners can rate the tutorial received.
- Learners can report any misconduct from the tutors.

2.3 User Classes and Characteristics

- Typical users such as students, who want the KFUPM tutoring software to help them exceed in their academic careers (Learners).
- Advanced/professional users, such as master students or higher studies, who want to teach other students or even benefit themselves (Tutors).
- Administrators.
- Software developers.

2.4 Operating Environment

The application will be developed in IOS and Android mobile systems.

2.5 Design and Implementation Constraints

- The source code must be written in Python for backend.
- The source code must be written in Dart for frontend.
- The application must use Flask for backend.
- The application must use Flutter for front end.
- The language of the app must be English.
- User may access from any device that has internet browsing capabilities and internet connection.
- The tutoring management system shall be running 24/7.

2.6 User Documentation

User documentation components give insights into the software and how to deal with it properly, therefore our tutoring software will provide a short introduction tutorial to help understand the software and use its functionality to the fullest. Our KFUPM tutoring software will also provide an installation and setup manual to guide intended users throughout setup. User documentation formats in IEEE include 3 main formats:

- Description Document
- Installation and Setup
- Product/user manual

2.7 Assumptions and Dependencies

We assume the following:

- The software has the ability to establish a connection required to access features. For example, users need internet to join a meeting but do not require internet to scroll through conversations or open files.
- The software is programmable and provide sufficient performances
- Human resources are available
- Budget availability
- Scheduling accuracy
- Scope doesn't change

With the following dependencies and constraints:

- Must finish the project strictly within the budget
- Must work with only available resources.
- The project should be ready for use within 4 months.

3. System Features

3.1 Creation of accounts for tutors and learners.

3.1.1 Description and Priority

Enabling the two types of end-users (tutors and learners) to create accounts. The account holder will have different features and different point of view of the application depending on his account type (tutor account or learner account).
The priority of this feature is High (8/10).

3.1.2 Functional Requirements

REQ-1: The system shall provide the ability for the all end-users to create an account.

3.2 Changing profile details

3.2.1 Description and Priority

This feature will enable the two types of end users (tutors and learners) to change their profile photo, updating their bios which consist their expertise, majors, age, etc...
The priority of this feature is Medium (5/10).

3.2.2 Functional Requirements

REQ-2: The system shall access the end user device data to change their profile information, including:

- Profile Photo
- Profile Biography, with a limit of 60 words.

3.3 Tutors can specify the tutoring subject/course, time, rate, type of tutoring

3.3.1 Description and Priority

This feature will enable tutors to create sessions, serving the main purpose of this application.
The priority of this feature is High (10/10).

3.3.2 Functional Requirements

REQ-3: Each tutor shall have a form to create sessions according to the following criteria:

- Subject/Course
- Time
- Rate
- Type (one to one, or one to many)
- Terms of use acceptance

3.4 Learners can search for available tutors/subjects/courses.

3.4.1 Description and Priority

This feature will enable learners to view tutoring sessions available for booking and choose the most appropriate session for them.
The priority of this feature is High (8/10).

3.4.2 Functional Requirements

REQ-4: The system shall provide the ability for the learner end user to view available sessions, with the following filters:

- Subject
- Time
- Rate
- Type

3.5 Learners can book a tutorial and pay in advance

3.5.1 Description and Priority

This feature will enable tutors to receive payments in advance and achieve the goal of this application to achieve a quality learning experience for learners.
The priority of this feature is High (9/10).

3.5.2 Functional Requirements

REQ-5: Each tutor shall have a payment form for booking tutoring sessions, payment shall include:

- Apple Pay
- STC Pay
- Credit card

3.6 Tutors can view registered students

3.6.1 Description and Priority

This feature will enable tutors to view registered students in their sessions.
The priority of this feature is High (9/10).

3.6.2 Functional Requirements

REQ-6: The system shall provide the tutor with the ability to view the students enrolled in his/her sessions.

3.7 Leaners can rate the tutorial Received

3.7.1 Description and Priority

This feature will enable leaners who completed the course to reflect their experience, quality of serves, their overall satisfaction. The rating will cover different element including quality of material, tutor rating.
The priority of this feature is Medium (5/10).

3.7.2 Functional Requirements

REQ-7: The system shall send a survey for leaners that include four sections and shall record results in the ratings database.

- On scale from 1 to 10, for overall satisfaction level
- On scale from 1 to 10, for satisfaction of tutors
- On scale from 1 to 10, for general feedback
- On scale from 1 to 10, for quality of learning.

3.8 Leaners can report any misconduct from tutors

3.8.1 Description and Priority

This feature will enable learners who completed a tutorial and who experienced misconduct from the tutor to report that misconduct to application administration.
The priority of this feature is Medium (5/10).

3.8.2 Functional Requirements

REQ-8: The system shall provide the ability for the learner end user to report misconduct, according to the following process:

- A report form shall be available in the tutor's profile page.
- The form shall include the following prompts:
  - Date of misconduct
  - Type of misconduct (according to University's code of conduct)
  - Evidence (video, audio, or witnesses)
- Form results shall be emailed to tutoring@kfupm.edu.sa

3.9 Administrators can delete accounts with misconduct

3.9.1 Description and Priority

This feature will enable administrators who received misconduct reports (according to university regulation) to delete accounts with relevance.
The priority of this feature is Medium (5/10).

3.9.2 Functional Requirements

REQ-9: The system shall provide the ability for the administrator delete any learner or tutor account.

3.10 Administrators should verify accounts.

3.10.1 Description and Priority

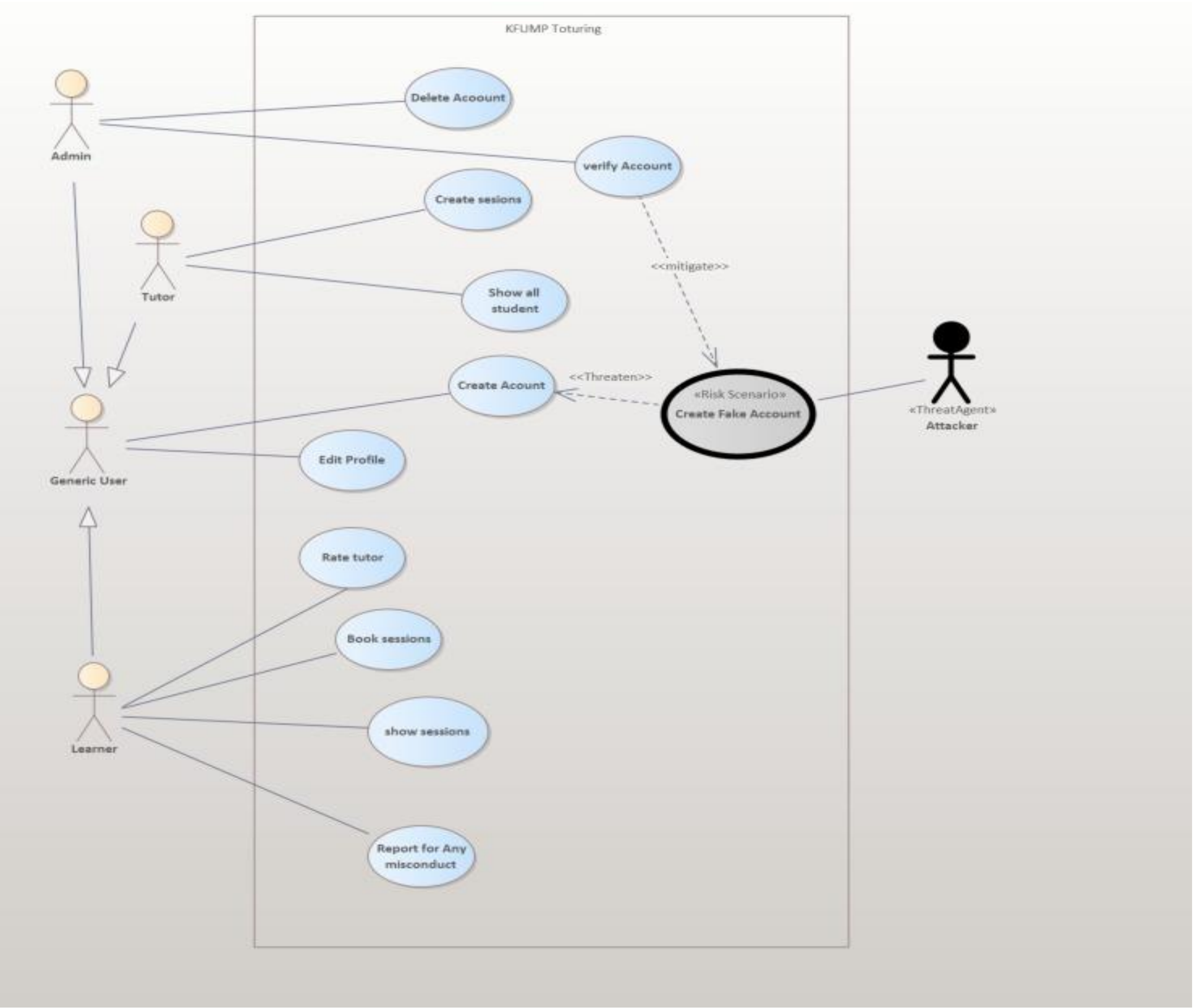
This feature will enable administrators to verify the accounts of newly registered users and their relative position, to ensure the integrity of the platform.
The priority of this feature is High (8/10).

3.10.2 Functional Requirements

REQ-10: The system shall provide the ability for the administrator verify any learner, tutor, or admin account.

4. Detailed Requirements

4.1 UML Use case Diagram



4.2 Functional Requirements

4.2.1 Requirement ID-01: Provide ability for the end-users to create an account

Priority	High
Effort	7 days
Risk	Low
Use Case(s)	UC-01
Description	Enabling the three types of end-users (admin, tutors, and learners) to create accounts. The account holder will have different features and different point of view of the application depending on his account type (tutor account or learner account or admin account).
Notes	N/A

4.2.1.1 UC ID-01: Account Creation

Use Case ID:	01
Use Case Name:	Account creation
Created By:	KHALID ALBAQAMI
Date Created:	4/13/2022
Actors:	End-users (admin, tutors, and learners)
Description:	Enabling the three types of end-users (admin, tutors, and learners) to create accounts. The account holder will have different features and different point of view of the application depending on his account type (tutor account or learner account or admin account).
Trigger:	End-user clicks on (Create an account) button.
Preconditions:	N/A
Post conditions:	The end user is given an account which suit his need depending on rather he is a tutor, leaner, or admin.
Normal Flow:	1-**Trigger**: The end-user clicks on (Create an account) button. 2-The end-user fills the required information in the given empty boxes (KFUPM email, phone number, and set a password). 3-The end-user selects from the given three choices (I am a learner, I am a tutor, I am an admin). 4-The end-user clicks on the (Done) button. 5-The accounts gets added to the verification page.
Alternative Flows:	-
Dependencies (Includes and extends relationships)	-
Frequency of Use:	Low
Threats	Fake account creation

4.2.2 Requirement ID-02: Provide ability for the end-users to edit profile information.

Priority	Medium
Effort	15 days
Risk	Medium
Use Case(s)	UC-02

Description	This use case will enable the two types of end users (tutors and learners) to change their profile photo, updating their bios which consists their expertise, majors, age, etc…
Notes	N/A

4.2.2.1 UC ID-02: Profile Edit

Use Case ID:	02
Use Case Name:	Profile Edit
Created By:	TURKI ALZAHRANI
Date Created:	4/13/2022
Actors:	End-users (admin, tutors, and learners)
Description:	This use case will enable the two types of end users (tutors and learners) to change their profile photo, updating their bios which consist their expertise, majors, age, etc..
Trigger:	Actor should click on edit profile button.
Preconditions:	1-Actor should open the app. 2-Actor should have a verified account. 3-Actor able to edit his profile twice within 14 days.
Post conditions:	1-Actor will see change immediately. 2-The change will be visible to other actors within 60 second.
Normal Flow:	1-The actor should open the app. 2- he/she should enter to his/her profile. 3-click on edit profile button.
Alternative Flows:	From step 3: If the actor changes his/her profile twice within 14 days the system will not allowed for a third change and the app will take the user to the main page.
Dependencies (Includes and extends relationships)	N/A
Frequency of Use:	High

4.2.3 Requirement ID-03: Provide ability for tutors to create sessions.

Priority	High
Effort	9 days
Risk	Medium
Use Case(s)	UC-03
Description	This use case will allow the tutor to create sessions.
Notes	N/A

4.2.3.1 UC ID-03: Create Session

Use Case ID:	03
Use Case Name:	Create Sessions
Created By:	SULTAN ALMOAMMAR
Date Created:	4/13/2022
Actors:	Tutors
Description:	This use case will allow the tutor to create sessions.
Trigger:	Tutors should click the create session button on screen.

Preconditions:	1-Tutors should open the app. 2-Tutors should have a verified account and log in successfully. 3-Tutors can create sessions at any given time.
Post conditions:	1-Tutors should be able to end the session immediately. 2-Tutors can start a new session after closing the present one.
Normal Flow:	1-The tutor should open the app. 2-The tutor should log into his account successfully. 3-Click on “create session”. 4-Fill session creation form (Subject/Course, Time, Rate, Type (one to one, or one to many)). 5-Accept Terms of use.
Alternative Flows:	From step 2: The tutor will not gain access to his account and therefore will not be able to create session. From step 4: The user will receive an error message and shall repeat the form, if some inputs were missing. From step 5: The user will receive an error message and shall accept the terms of use, if he/she didn’t accept it.
Dependencies (Includes and extends relationships)	N/A
Frequency of Use:	High

4.2.4 Requirement ID-04: Provide ability for tutors to create sessions.

Priority	High
Effort	6 days
Risk	Low
Use Case(s)	UC-04
Description	This use case will allow the learner to show all available tutors.
Notes	N/A

4.2.4.1 UC ID-04: Show Sessions

Use Case ID:	04
Use Case Name:	Show Sessions
Created By:	ABDULLAH ALHABIB
Date Created:	4/13/2022
Actors:	Learners
Description:	This will enable learners to view available sessions for booking and choose the most appropriate courses for them.
Trigger:	Learners should click the show all sessions button on screen.
Preconditions:	1-Learners must open the app. 2-Learners must have a verified account and log in successfully.
Post conditions:	1-Learners should be able to look to the available sessions.
Normal Flow:	1-The learner should open the app. 2-The learner should log into his account successfully. 3-The learner may use filters with the following criteria: Subject, Time, Rate, and Type. 4-click on “Show all available sessions”. 5- choose appropriate course.
Alternative Flows:	From step 2: The learner will not gain access to his account and therefore will not be able to view sessions.

Dependencies (Includes and extends relationships)	N/A
Frequency of Use:	High

4.2.5 Requirement ID-05: Provide ability for learners to book sessions.

Priority	High
Effort	15 days
Risk	High
Use Case(s)	UC-05
Description	This use case will allow the learner to book and pay for sessions.
Notes	N/A

4.2.5.1 UC ID-05: Book Session

Use Case ID:	05
Use Case Name:	Book Session
Created By:	MUTAZ ALSHAHRANI
Date Created:	4/13/2022
Actors:	Learners
Description:	This will enable learners to book and pay for available sessions.
Trigger:	Learners should click the show all sessions button on screen and click the book session button besides the relevant session.
Preconditions:	1-Learners must open the app. 2-Learners must have a verified account and log in successfully.
Post conditions:	1-Learners should be registered for relevant session 2-Tutors should receive payment.
Normal Flow:	1-The learner should open the app. 2-The learner should log into his account successfully. 3-click on “Show all available sessions”. 4-The learner may use filters with the following criteria: Subject, Time, Rate, and Type. 5- Choose appropriate course. 6-Click Book button. 7-Select payment option (Apple Pay, STC Pay, Credit Card). 8-Proceed to payment website, with relevant API. 9- After success, user should be registered.
Alternative Flows:	From step 2: The learner will not gain access to his account and therefore will not be able to book sessions. From step 8: The learner will be returned to the application, with an error message, if the payment failed.
Dependencies (Includes and extends relationships)	N/A
Frequency of Use:	High

4.2.6 Requirement ID-06: Provide ability for tutor to view students.

Priority	High
Effort	15 days

Risk	Low
Use Case(s)	UC-06
Description	This use case will allow the tutor to view enrolled students.
Notes	N/A

4.2.6.1 UC ID-06: View Students

Use Case ID:	06
Use Case Name:	View Students
Created By:	SAUD ALFARIS
Date Created:	4/13/2022
Actors:	Tutors
Description:	This will enable tutors to view students enrolled in their sessions.
Trigger:	Tutor should click the show students in the respective session page.
Preconditions:	1-Tutor must open the app. 2-Tutor must have a verified account and log in successfully. 3-Tutor must have a created session.
Post conditions:	1-A list of enrolled students will be shown to the tutor.
Normal Flow:	1-The tutor should open the app. 2-The tutor should log into his account successfully. 3-Click on respective session page. 4-Click “view students” button
Alternative Flows:	From step 2: The tutor will not gain access to his account and therefore will not be able to book sessions.
Dependencies (Includes and extends relationships)	N/A
Frequency of Use:	High

4.2.7 Requirement ID-06: Provide ability for learners to rate tutors.

Priority	Medium
Effort	15 days
Risk	Medium
Use Case(s)	UC-06
Description	This use case will allow the learner to rate according to their sessions.
Notes	N/A

4.2.7.1 UC ID-06: Rate Tutor

Use Case ID:	06
Use Case Name:	Rate Tutor
Created By:	TURKI ALZAHRANI
Date Created:	4/13/2022
Actors:	Learners
Description:	This feature will enable leaners who completed the course to reflect their experience, quality of serves, their overall satisfaction. The rating will cover different element including quality of material, tutor rating. The rating will be from 0-10.

Trigger:	Learners should click the rate tutor on the session page.
Preconditions:	1-Learners must open the app. 2-Learners must have a verified account and log in successfully.
Post conditions:	1-Rate tutor button will disappear. 2-Rating will be recorded in the ratings database.
Normal Flow:	1-The learner should open the app. 2-The learner should log into his account successfully. 3-The learner should open the session page. 4-The learner should click the "Rate" button. 5-The learner should fill the rating form, with ratings according to the following criteria: - On scale from 1 to 10, for overall satisfaction level - On scale from 1 to 10, for satisfaction of tutors - On scale from 1 to 10, for general feedback - On scale from 1 to 10, for quality of learning. 6- The learner should click the submit button. 7-Ratings should be submitted to the ratings database.
Alternative Flows:	From step 2: The learner will not gain access to his account and therefore will not be able to book sessions. From step 5: The learner shall receive an error message and shall repeat the form if some inputs were missing.
Dependencies (Includes and extends relationships)	N/A
Frequency of Use:	High

4.2.8 Requirement ID-07: Report Misconduct.

Priority	High
Effort	7 days
Risk	High
Use Case(s)	UC-07
Description	This use case will allow the learners to report misconduct from tutors.
Notes	N/A

4.2.8.1 UC ID-07: Report Misconduct

Use Case ID:	07
Use Case Name:	Report Misconduct
Created By:	SULTAN ALMOAMMAR
Date Created:	4/13/2022
Actors:	Learners
Description:	This use case will allow the learners to report misconduct from tutors.
Trigger:	Clicking the "Report" button in the session page
Preconditions:	1-Learner should open the app. 2-Learner should have a verified account and log in successfully. 3-Learner should have a book session with the reported tutor.
Post conditions:	1-Misconduct report will be emailed to administration email.
Normal Flow:	1-The learner should open the app. 2-The learner should log into his account successfully. 3-The learner should open the session page.

	4-The learner clock the “Report” Button. 5-The learner should fill the report form containing the following criteria: - Date of misconduct - Type of misconduct (according to University’s code of conduct) - Evidence (video, audio, or witnesses) 6- The report should be sent to the following email: tutoring@kfupm.edu.sa
Alternative Flows:	From step 2: The learner will not gain access to his account and therefore will not be able to book sessions. From step 5: The learner shall receive an error message and shall repeat the form if some inputs were missing.
Dependencies (Includes and extends relationships)	N/A
Frequency of Use:	Low

4.2.9 Requirement ID-08: Delete Account.

Priority	High
Effort	20 days
Risk	Medium
Use Case(s)	UC-08
Description	This use case will allow the administrators to delete misconducted tutors.
Notes	N/A

4.2.9.1 UC ID-08: Delete Account

Use Case ID:	08
Use Case Name:	Delete Account
Created By:	TURKI ALZAHRANI
Date Created:	4/13/2022
Actors:	Administrators
Description:	This use case will allow the administrators to delete misconducted tutors.
Trigger:	Receiving a valid misconduct report.
Preconditions:	1-Admin should open the app. 2-Admin should have a verified account and log in successfully.
Post conditions:	1-Relevant tutor will be deleted.
Normal Flow:	1-The admin should open the app. 2-The admin should log into his account successfully. 3-The admin should open the delete account page. 4-The admin should enter the tutor id. 5-The admin should click the delete button.
Alternative Flows:	From step 2: The learner will not gain access to his account and therefore will not be able to book sessions. From step 5: The admin shall receive an error message and should repeat the process if the user account did not exist
Dependencies (Includes and extends relationships)	N/A
Frequency of Use:	Low

4.2.10 Requirement ID-09: Verify Account.

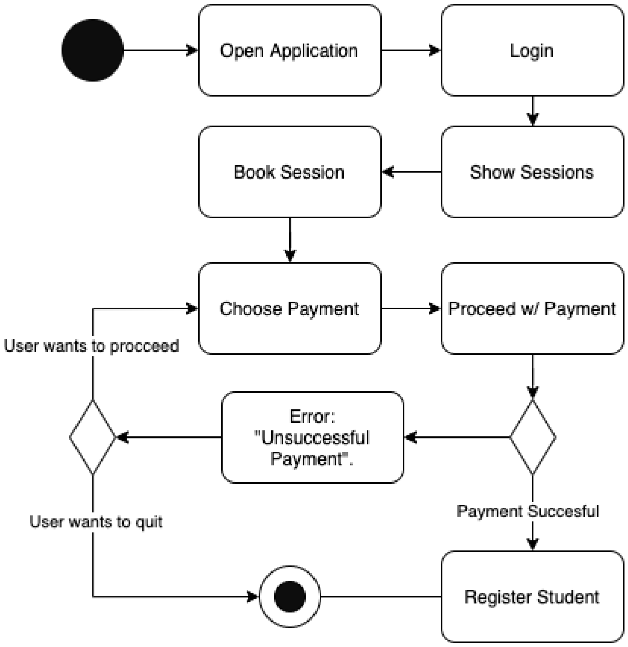
Priority	High
Effort	7 days
Risk	High
Use Case(s)	UC-09
Description	This use case will allow the administrators to verify created accounts.
Notes	N/A

4.2.10.1 UC ID-09: Verify Account

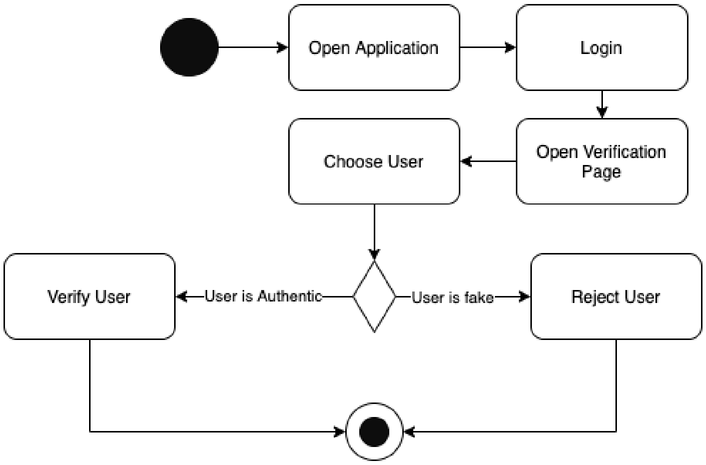
Use Case ID:	09
Use Case Name:	Verify Account
Created By:	SULTAN ALMOAMMAR
Date Created:	4/13/2022
Actors:	Administrators
Description:	This use case will allow the administrators to verify created tutors.
Trigger:	The creation of a new account.
Preconditions:	1-Admin should open the app. 2-Admin should have a verified account and log in successfully. 3-A new account must be created.
Post conditions:	1-New account will be verified and have all relevant permissions.
Normal Flow:	1-The admin should open the app. 2-The admin should log into his account successfully. 3-The admin should open the verification page. 4-The admin should choose the verified user. 5-The admin should click the verify button.
Alternative Flows:	From step 2: The learner will not gain access to his account and therefore will not be able to book sessions. From step 5: If the user is fake, admin can choose the reject button.
Dependencies (Includes and extends relationships)	N/A
Frequency of Use:	Low
Mitigates	Creation of fake accounts.

4.3 UML Activity Diagrams

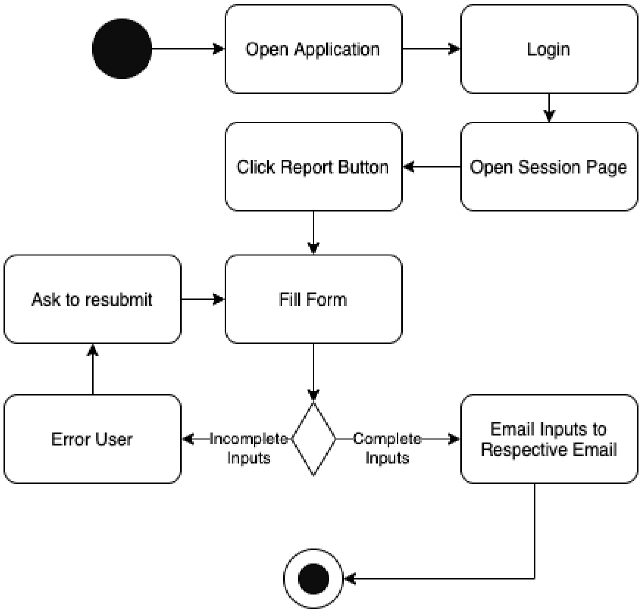
Booking Sessions



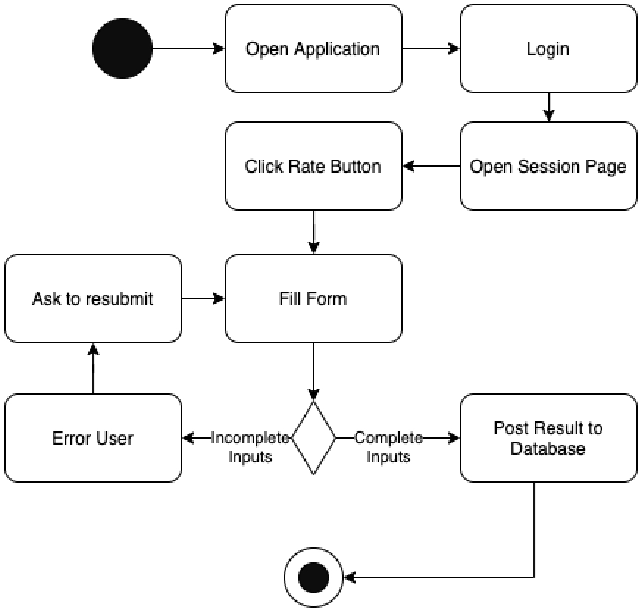
Verifying Users



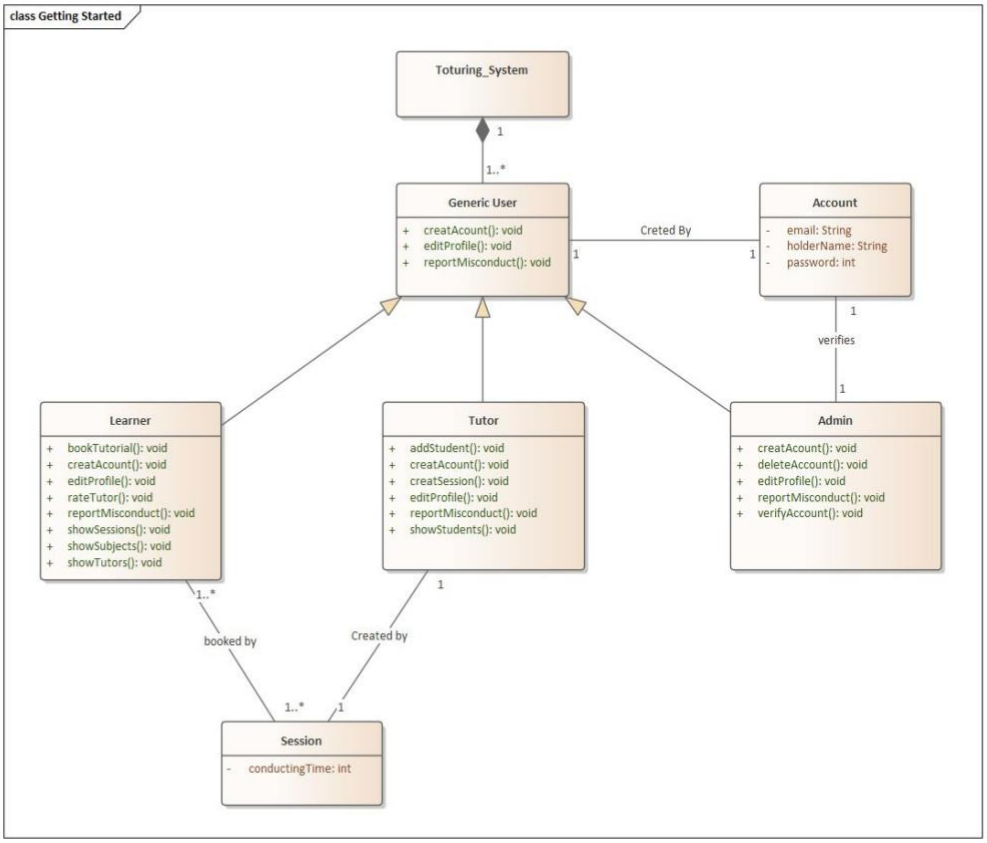
Report Tutor



Rate Tutor



4.4 System Domain Model



4.5 Non-functional Requirements

4.5.1 Performance Requirements

REQ10: The system shall display changes to any profile information within a maximum of 60 seconds.

4.5.2 Security Requirements

REQ11: All created accounts must be verified for using the application, with the exception of the first registered administrator.

4.5.3 Other Software Quality Attributes

REQ12: The system's iOS interface shall be consistent with "Apple's Human Interface Guidelines".
REQ13: The system's Android interface shall be consistent with "Android's Quality Guidelines".

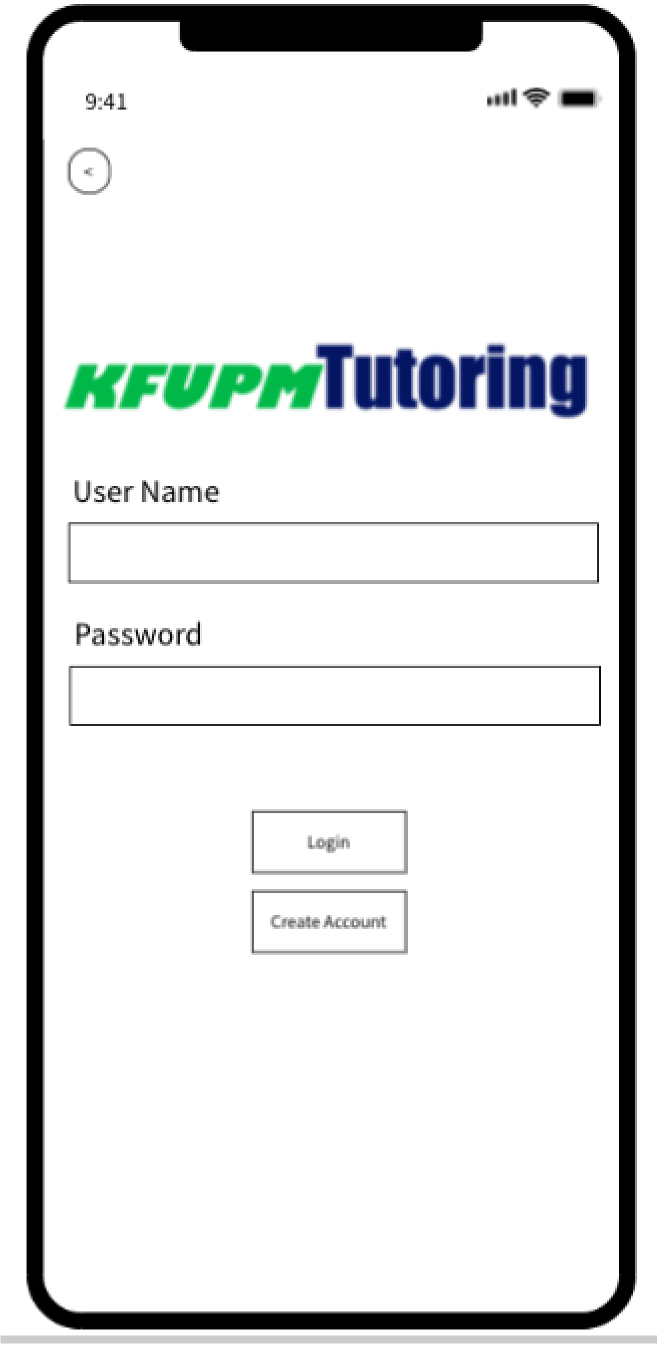
4.5.4 Other Requirements

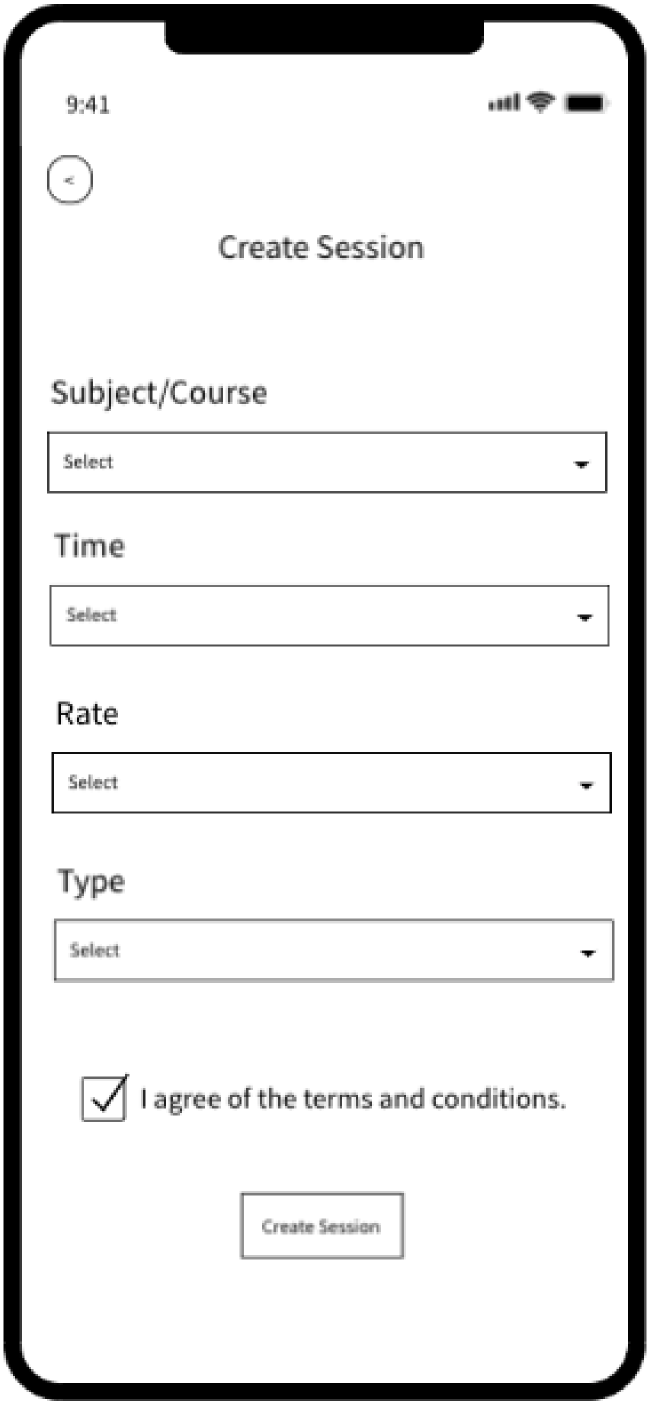
REQ14: All system databases must be installed in KFUPM's private servers.
REQ15: Reporting must be compatible with KFUPM's Code of Conduct.

5. External Interface Requirements

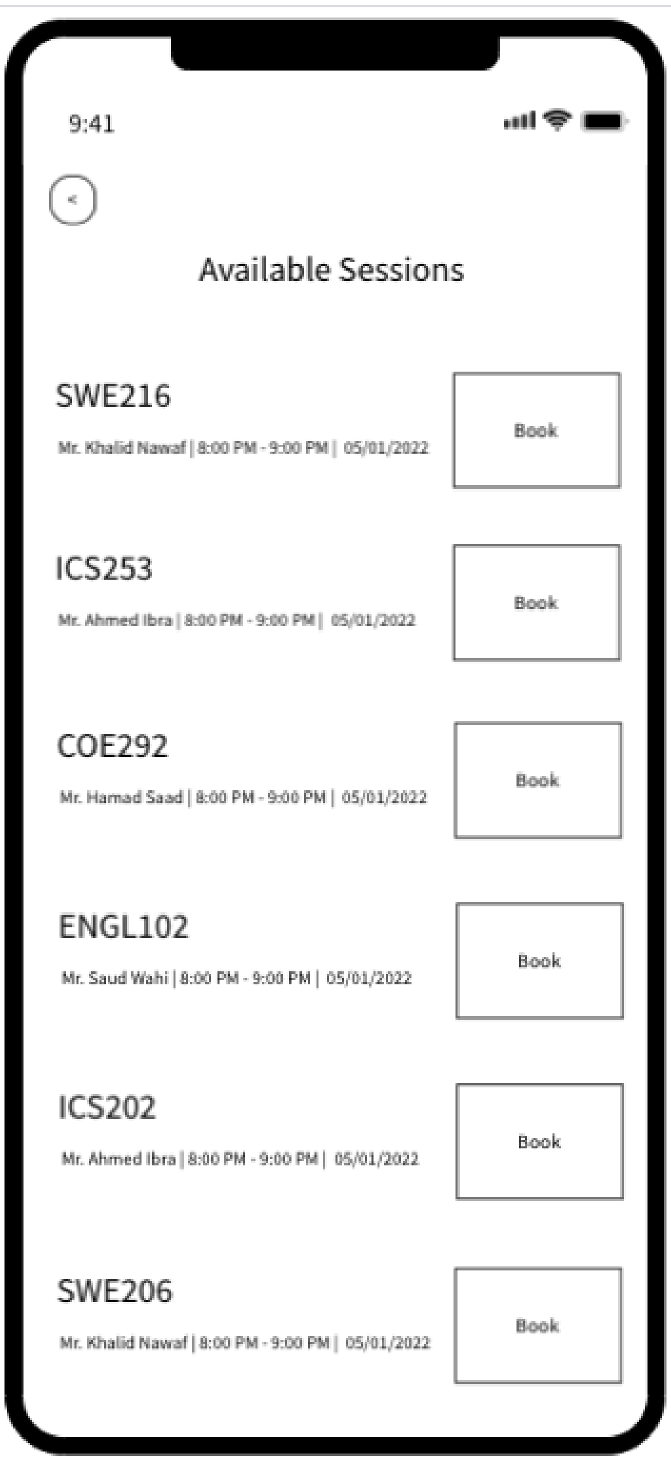
5.1 User Interfaces

To demonstrate the required interface for this project, the following simulation was designed:





☑ I agree of the terms and conditions.



Available Sessions

SWE216
Mr. Khalid Nawaf | 8:00 PM - 9:00 PM | 05/01/2022

ICS253
Mr. Ahmed Ibra | 8:00 PM - 9:00 PM | 05/01/2022

COE292
Mr. Hamad Saad | 8:00 PM - 9:00 PM | 05/01/2022

ENGL102
Mr. Saud Wahi | 8:00 PM - 9:00 PM | 05/01/2022

ICS202
Mr. Ahmed Ibra | 8:00 PM - 9:00 PM | 05/01/2022

SWE206
Mr. Khalid Nawaf | 8:00 PM - 9:00 PM | 05/01/2022

5.2 Hardware Interfaces

- The project's application must be compatible with the following devices:

  - Apple iPhone
  - Apple iPad
  - Samsung Galaxy
  - Google Pixel
  - Samsung OnePlus

- The application server-side computations must be performed on KFUPM ITC servers.

5.3 Software Interfaces

- The application must be compatible with iOS version 15 operating system.
- The application must be compatible with Android version 12 operating system.
- The application must use Dart as the front-end development language.
- The application must use Flutter as the front-end development library.
- The application must use Python for the backend system.
- The application must use Flask as the back-end development library.
- The application must use SQLite as the database development language.
- The application must use Oracle's Databases for the database environment.

5.4 Communications Interfaces

5.4.1 Protocol Requirements

- The application must use the encrypted HTTPS protocol for communications.
- The application must use HTTPS standard response codes, given the following:
  - Informational responses (100-199)
  - Successful Responses (200-299)
  - Redirection Responses (300-399)
  - Client Error Responses (400-499)
  - Server Error Responses (500-599)

5.4.2 Email Requirements

Considering system functional requirements and the nature of system processes, email and stable internet connectivity are essential system components of the overall system architecture. The KFUPM tutoring system will use SMTP email protocol.

6. Appendix A: Glossary

Acronyms used in this document:

- KFUPM: King Fahd University of Petroleum and Minerals.
- SRS: Software Requirements Specification.

- IEEE: Institute of Electrical and Electronics Engineers.
- REQ: Requirement.
- UML: Unified Modeling Language.
- UC: Use Case.

7. Appendix B: Issues List

Issues not yet resolved:

- Integration with the KFUPM Student Success Center.
- University President Approval.
- Agreement with KFUPM ITC for private server usage.
- Purchase of Oracle database services.