College of Computer & Information Sciences

Software Engineering Department

SWE321 - مستجد

Spring 2022

Software design & architecture project

مستجد

Section : 49393 | group #3

Student name	ID
Reem alshareef (leader)	**441200087**
Renad Alsubaie	**441200868**
Monerah Alsubaie	**441200880**
Fai Alotaibi	**441201811**
Jouri Alanazi	**441200780**

Supervised by : Dr.abeer najar

Table of Content

Revision Control History:

Version Number	Date	Description of change	Reason for change
1	24 September , 2022	First phase of the project	——
2	1 November, 2022	Second phase of the project	——
3	1 November, 2022	Second phase of the project	——
4	3 November, 2022	Third phase of the project	——

Introduction:

As system designers and architects, our job during this phase was to define the methodology for the system as well as a high-level view of our system architecture, which included modeling the class, use case, and sequence diagrams as well as defining the architecture of the system and any information pertaining to the database and storage methods we will be using. We also briefly discussed the quality attributes and the quality assurance of our system. Finally, we discussed our considerations for the future and our plans for system upgrades and improvements.

Problem Domain Analysis:

A system that puts in your hands everything you need to know about the university education system within the Kingdom of Saudi Arabia. colleges. specialties. General percentages of admission to colleges. Which helps you to determine which college or major you can join and the study plan for each major, as well as comprehensive information about each major

تطبيق جامعي

The goal of the application is to help the student achieve success in his university career; If you are a university student and need a program that helps you organize your time and view the academic calendar

تطبيق ضوء الخريج

An application that shows you many specialties in a variety of fields. It presents an overview of the specialization, its number of years, acceptance rates and job opportunities. The application also supports the calculation of the weighted percentage.

موقع التوجيه الجامعي

The site sought to serve high school graduates , by providing a variety of services: View a group of universities and specializations available.,Academic entry requirements and living needs.,A comprehensive guide on student guidance for high school graduates for the various universities available.,Taking into account the needs of the labor market and urging students to choose the required specializations.provid Controls and procedures of the Ministry of Education and developments in matters.

A site for all Saudi university students, the idea is that it contains all the duties and tests, as well as references to most of the subjects in the past years that you can afford and benefit from in your university studies

MSTAGED It aims to assist every new student in their college career and provide them with complete knowledge about everything they are interested in whether they are interested in majors, used books, study resources, transfers, major change, and important dates.

	Direct communication with students	**Provide freshmen with study resources and places of sale of this resources**	**Provide them with numbers and accounts for transportation**	**Weighted calculation, allocation, and acceptance ratios**	**Informing them of the services provided by the university, including the possibility of studying, restaurants, and facilities**	**Students will be informed of important dates, updated policies, changes in key procedures, and the syllabus**	**Providing information on specializations with admission requirements and their functions**
	✓	✓	✓	✓	✓	✓	✓

دليل الخريج				✓			✓
موقع التوجيه الجامعي موقع التوجيه الجامعي						✓	✓
Bazinga!		✓					
ضوء الخريج تطبيق ضوء الخريج				✓			✓
تطبيق جامعي						✓	

The System Context View:

User scenario A: Tack advice

The user opens the ***MSTAGED*** app, ***MSTAGED*** home page will appear
the user selects the "Tack advice" option. After that the user selected "Search" option, the user should Input specialization name, The user will see who is available to talk and she/he will Start Text chat to get advice from experienced students.

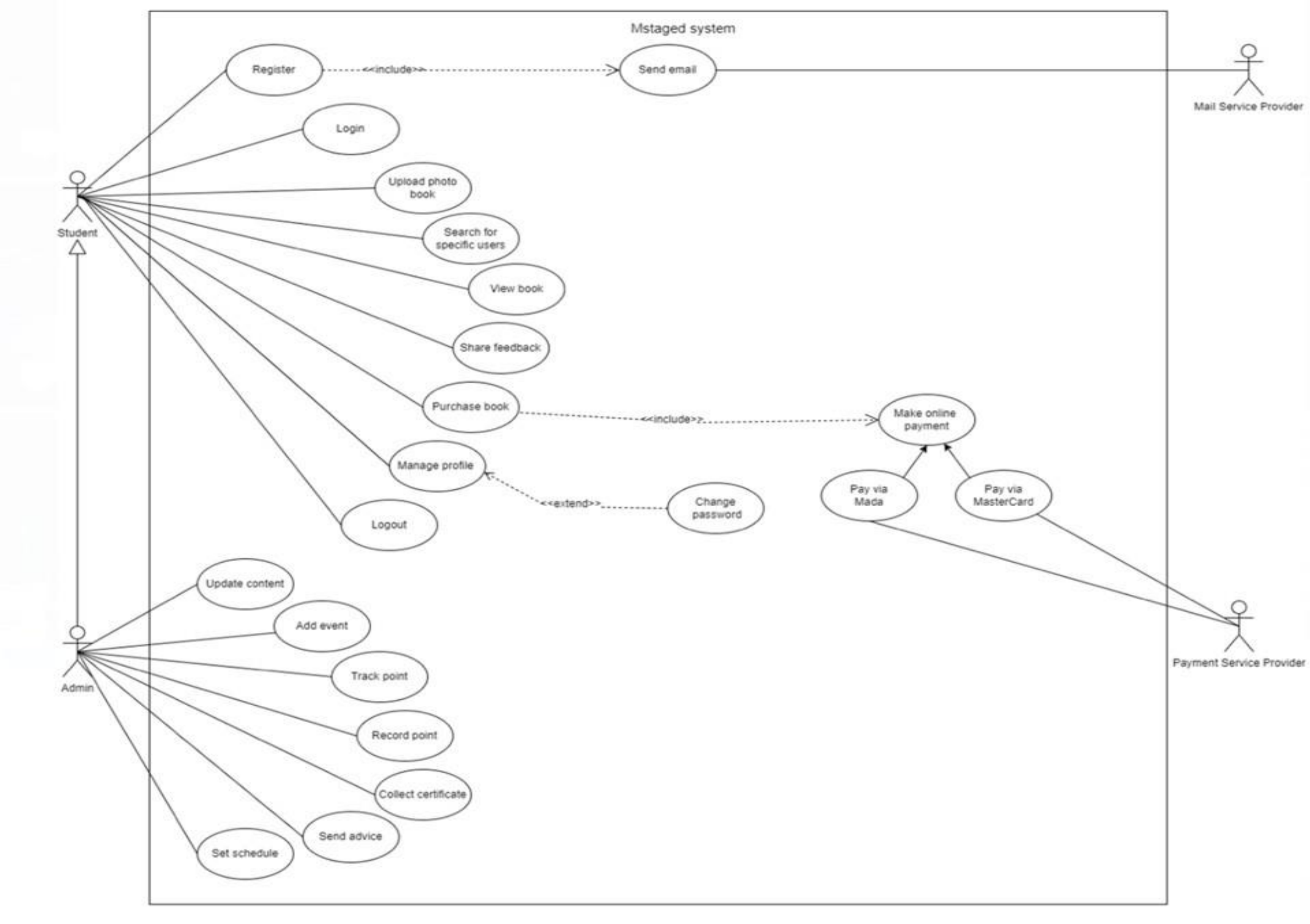
User scenario B: Book exchange

The user opens the ***MSTAGED*** app, ***MSTAGED*** home page will appear
the user selects the "Share book" option. After that, the user can take a picture for a book or choose a book picture from his/her album, adding name of book and selects "Share " option
The system will publish the book to other users which have same major .System

Requirements :

Functional Requirement (FR)

Mstaged's Use case diagram.



Non-Functional Requirement (NFR)

Portability
5.2.1 The system shall be able to run in all operating systems environments.

Scalability
5.2.2 The system shall be able to support an annual growth of 30% of new students.

Security
5.2.3 Password shall be required to access student screens.

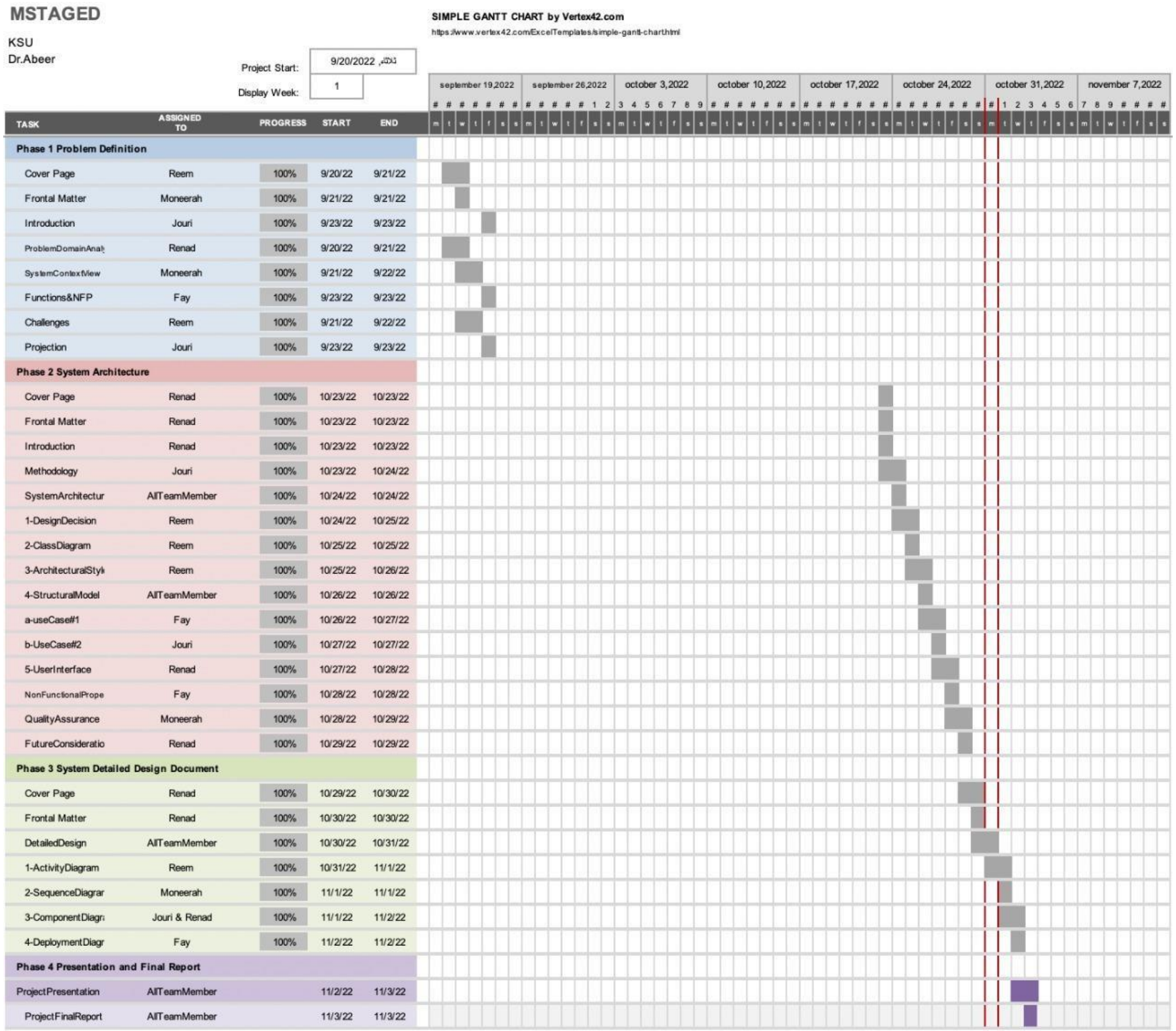
Usability
5.2.4 The new user shall be able to learn all system functionality in less than 20 minutes.

Performance
5.2.5 Average system response time time should be less than four seconds.
5.2.6 The system shall accommodate 3000 concurrent users.

Challenges :

We will face a lot of challenges in our project , we have to deliver it in small period of time, we have to find an appropriate architecture for our application , we need to meet many times to discuss and manage our tasks.

Projection :



MSTAGED

KSU
Dr.Abeer

Project Start: 9/20/2022 ،ثلاثاء
Display Week: 1

SIMPLE GANTT CHART by Vertex42.com
https://www.vertex42.com/ExcelTemplates/simple-gantt-chart.html

TASK	ASSIGNED TO	PROGRESS	START	END
Phase 1 Problem Definition				
Cover Page	Reem	100%	9/20/22	9/21/22
Frontal Matter	Moneerah	100%	9/21/22	9/21/22
Introduction	Jouri	100%	9/23/22	9/23/22
ProblemDomainAnal	Renad	100%	9/20/22	9/21/22
SystemContextView	Moneerah	100%	9/21/22	9/22/22
Functions&NFP	Fay	100%	9/23/22	9/23/22
Challenges	Reem	100%	9/21/22	9/22/22
Projection	Jouri	100%	9/23/22	9/23/22
Phase 2 System Architecture				
Cover Page	Renad	100%	10/23/22	10/23/22
Frontal Matter	Renad	100%	10/23/22	10/23/22
Introduction	Renad	100%	10/23/22	10/23/22
Methodology	Jouri	100%	10/23/22	10/24/22
SystemArchitectur	AllTeamMember	100%	10/24/22	10/24/22
1-DesignDecision	Reem	100%	10/24/22	10/25/22
2-ClassDiagram	Reem	100%	10/25/22	10/25/22
3-ArchitecturalSty	Reem	100%	10/25/22	10/26/22
4-StructuralModel	AllTeamMember	100%	10/26/22	10/26/22
a-useCase#1	Fay	100%	10/26/22	10/27/22
b-UseCase#2	Jouri	100%	10/27/22	10/27/22
5-UserInterface	Renad	100%	10/27/22	10/28/22
NonFunctionalPrope	Fay	100%	10/28/22	10/28/22
QualityAssurance	Moneerah	100%	10/28/22	10/29/22
FutureConsideratio	Renad	100%	10/29/22	10/29/22
Phase 3 System Detailed Design Document				
Cover Page	Renad	100%	10/29/22	10/30/22
Frontal Matter	Renad	100%	10/30/22	10/30/22
DetailedDesign	AllTeamMember	100%	10/30/22	10/31/22
1-ActivityDiagram	Reem	100%	10/31/22	11/1/22
2-SequenceDiagrar	Moneerah	100%	11/1/22	11/1/22
3-ComponentDiagr	Jouri & Renad	100%	11/1/22	11/2/22
4-DeploymentDiagr	Fay	100%	11/2/22	11/2/22
Phase 4 Presentation and Final Report				
ProjectPresentation	AllTeamMember		11/2/22	11/3/22
ProjectFinalReport	AllTeamMember		11/3/22	11/3/22

Methodology:

The suitable methodology for the **MSTAGED** system to adopt is the Agile Methodology. The Agile Methodology proposes an incremental and iterative approach to software design that breaks the system into individual modules and builds a detailed plan for only a small module of the entire system.

The principle of Agile Methodology will allow us to manage user expectations since the changes are flexible and can be done by the user at any phase, providing us with high-quality products in a short period of time and at a low cost.
Since our System considered a complex and distributed system which needs to manage a huge amount of data with rich functionality we planned to use **MVC** distributed as 3-tier Architecture.
Our system will consist of four modules/subsystems: update content, Purchase Book, Share Feedback, and Set Schedule.

However, our system will be faced with some risks since we are using Agile methodology One of the main risks is **Poor resource planning** Because Agile is based on the idea and it’s challenging to predict efforts like cost and time and resources required at the beginning of a project, **Difficult measurement** Since Agile delivers in increments and tracking progress requires you to look across cycles.

Modules/Subsystem	Responsibilities	Duration	Prototype Demonstrations	Design Deadline	Final Implementation	Final Demonstrations
1-Update Content	All Team	6 weeks	10 August, 2022	11 August	19 August	5 September
Analysis #1	Requirement Team	2 weeks		13 August	21 August	10 September
Design #1	Design Team	1 week		15 August	24 August	15 September
Implementation #1	Programmers Team	2 weeks		16 August	27 August	20 September
Unit Testing #1	Testing Team	1 week		17 August	30 August	25 September
2-Purchase Book	All Team	6 Weeks	20 September, 2022	21 September	1 October	11 October
Analysis #2	Requirement Team	2 weeks		23 September	3 October	12 October
Design #2	Design Team	1 week		25 September	5 October	13 October
Implementation #2	Programmers Team	2 weeks		27 September	7 October	15 October
Integration Testing #2	Integration Management Team	1 week		29 September	9 October	17 October
3-Share Feedback	All Team	6 Weeks	18 October, 2022	19 October	29 October	8 November
Analysis #3	Requirement Team	2 weeks		21 October	31 October	10 November
Design #3	Design Team	1 week		23 October	2 November	12 November
Implementation #3	Programmers Team	2 weeks		25 October	4 November	14 November
Integration Testing #3	Integration Management Team	1 week		27 October	6 November	16 November
4-Set Schedule	All Team	8 Weeks	17 November, 2022	18 November	28 November	8 December
Analysis #4	Requirement Team	3 weeks		20 November	30 November	10 December
Design #4	Design Team	1 week		22 November	2 December	12 December
Implementation #4	Programmers Team	3 weeks		24 November	4 December	14 December
Integration Testing #4	Integration Management Team	1 week		26 November	6 December	16 December

Design decisions:

Since our project is considered a complex, distributed system used by many distributed users that needs to manage a huge amount of data with rich functionality, we decide to use a multi-tier architecture design for many purposes.

First of all, the middle tire in a multi-tire system is an important part of the enhancement of reusability and scalability and reducing the traffic on the client side, so we will not suffer from fat clients. Our system will have many improvements and releases in the future, so it will be easy to modify and add new functionality since any business related changes are made only in the middle tire.

Use Case Description

1. Use Case Share feedback

Use Case Description	
System: MSTAGED System	
Use Case name: Share feedback	
Primary actor: Student	**Secondary actor(s):** None
Description: This use case describes how a student share new feedback.	
Relationships ● **Includes:** None ● **Extends:** None	
Pre-conditions: Student is successfully logged in.	
Steps	
Primary Actor (Student)	**System**
1. The use case begins when the student selects “Share feedback”.	
	2. The system shows a text box to type the feedback.
3. The student fills the text box with the feedback.	
4. The student submits the feedback.	
	5. The system will add the new feedback.
	6. The system presents a message indicating that the new feedback was added successfully.
Alternative and exceptional flows: ● **Missing Field / Incorrect Format:** If in step 3 the student misses a required field or enters a field in incorrect format, then: 1. The system will present a message indicating required fields or format. 2. Step 2 is resumed. • **Student Quits:** If at any step before step 4 the student selects “cancel”: 1. The use case will end with a failure condition.	

Post-conditions: • **Successful Condition:** The new feedback is successfully added to the system. • **Failure Condition:** The system fails to add the new feedback.

2. Use Case Purchase Book

Use Case Description	
System: MSTAGED	
Use Case name: Purchase Book	
Primary actor: Student	**Secondary actor(s):** none
Description: This use case describes how a student can Purchase Book	
Relationships • **Includes:** none. • **Extends:** None.	
Pre-conditions: -The student has Logged-In Successfully.	

Primary Actor (Student)	**System (MSTAGED)**	**Secondary Actor(s) (bank system)**
1.The student selects “ Purchase Book ”.		
	2.The system shows two option to pay via Mada or via Mater Card.	
3.The student select one of the options.		
	4.The system displays payment form.	
5. The Student fills the form with the payment form with the following: -Card Holder Name -Card Number -CVV -Expiration Date.		

6. The Student Submit the form.	7. The system directs the student payment form to the bank system 9.The system displays that the book has been purchased successfully.	8.bank system make the payment
Alternative and exceptional flows: **in step 8 if the validation step fails, then:** **1.The system displays a message including invalid input.** **2.Step 5 is resumed.**		
Post-conditions: **Successful condition**: The Student Purchase the Book successfully. **Failure condition:** The Student fill wrong CVV so the payment fail .		

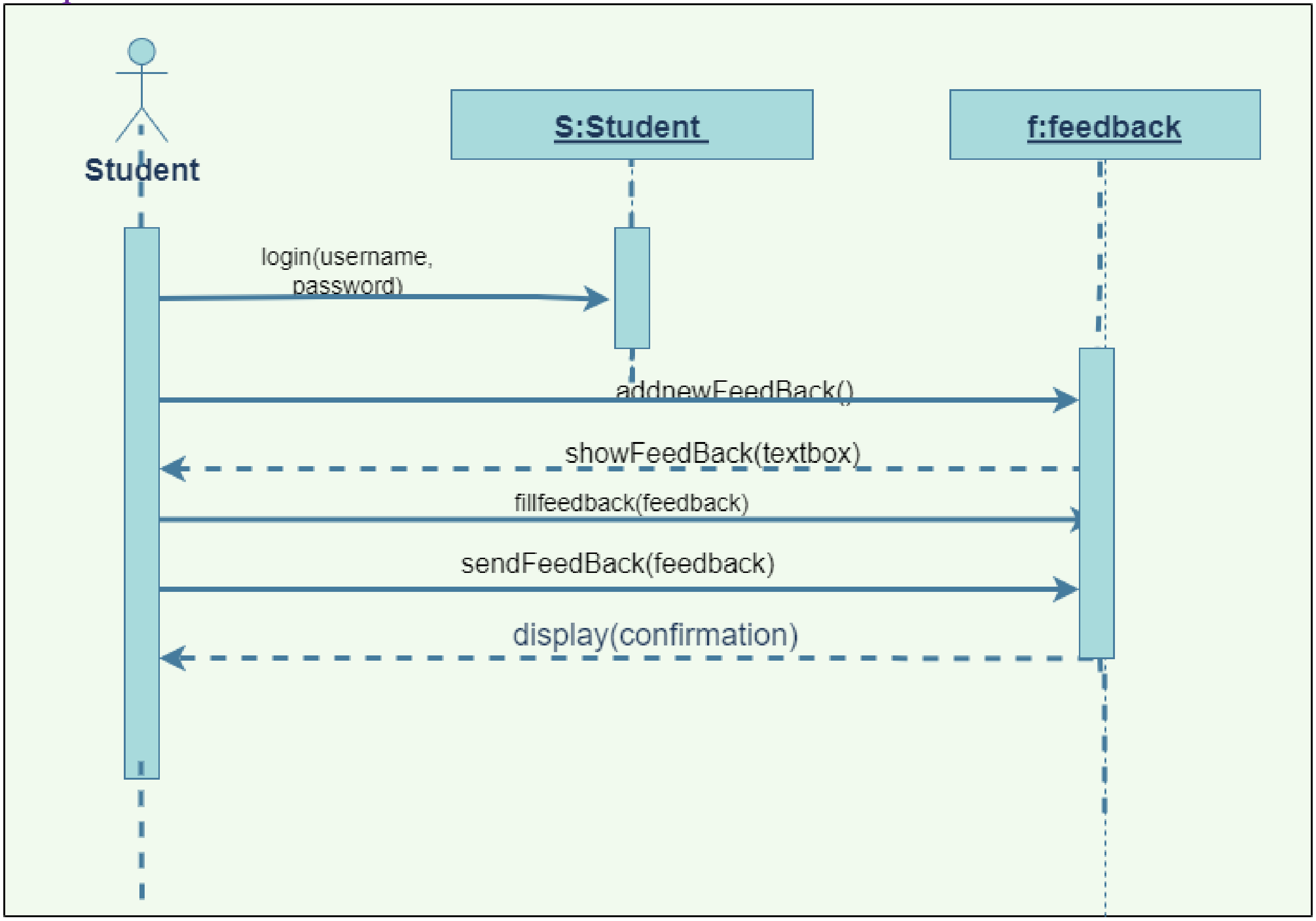
3. Use Case add book for sale

Use Case Description	
System: MSTAGED System	
Use Case name: Share feedback	
Primary actor: admin	**Secondary actor(s):** None
Description: This use case describes how an admin insert new book for sale.	
Relationships ● **Includes:** None ● **Extends:** None	
Pre-conditions: admin is successfully logged in.	
Steps	
Primary Actor (admin)	**System**

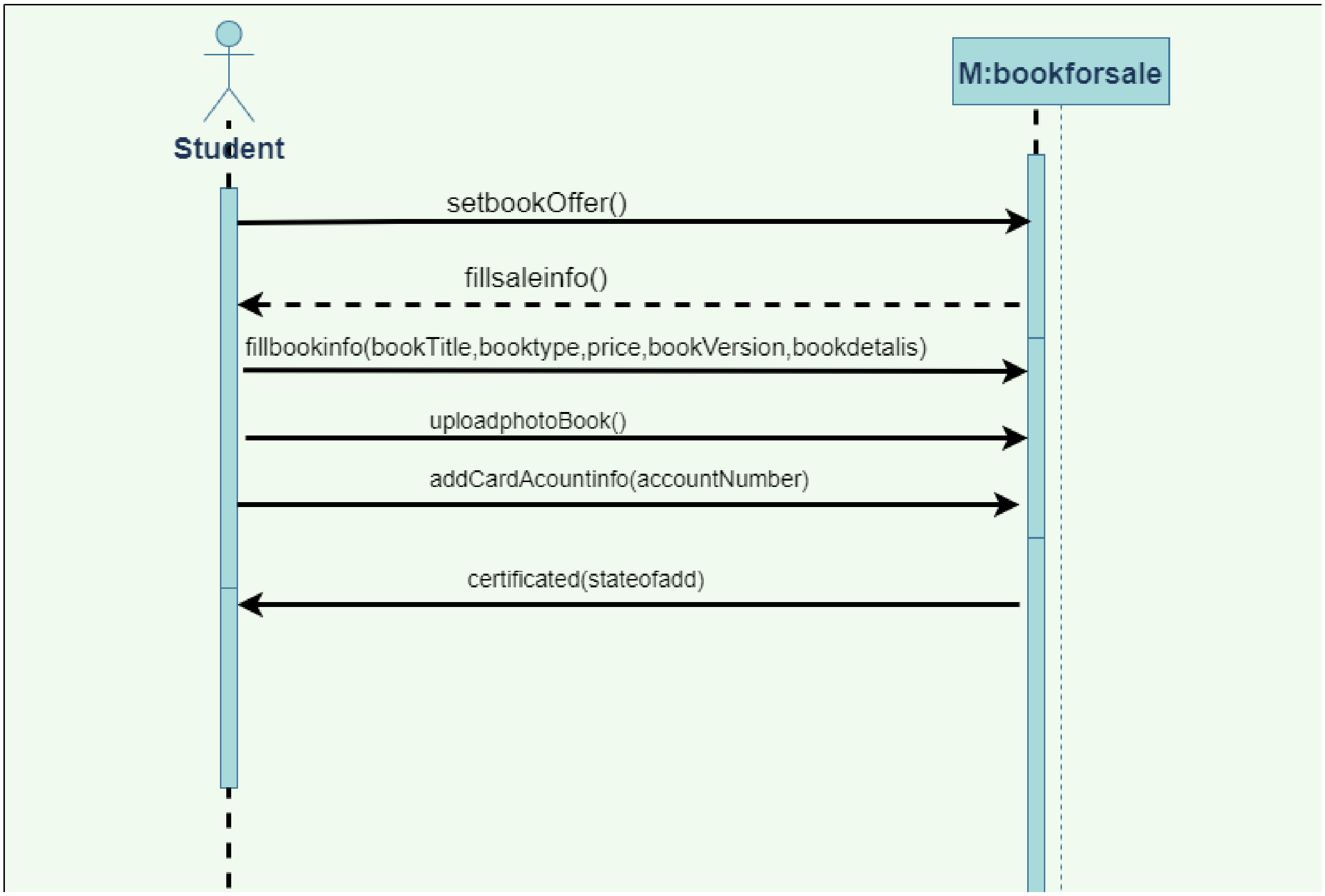
<table>
<tr><td>1. The use case begins when the student selects “add book for sale ”.
3. The admin fills the text box with the sale book info
4. The student submits the sale informarion.</td><td>2. The system shows a text box to write the sale book information.
5. The system will add the new sale offer.
6. The system presents a message indicating that the new sale offer was added successfully.</td></tr>
<tr><td colspan="2"><b>Alternative and exceptional flows:</b>
● <b>Missing Field / Incorrect Format:</b>
If in step 3 the student misses a required field or enters a field in incorrect format, then:
1. The system will present a message indicating required fields or format.
2. Step 2 is resumed.
● <b>Student Quits:</b>
If at any step before step 4 the student selects “cancel”:
1. The use case will end without add the sale offer.</td></tr>
<tr><td colspan="2"><b>Post-conditions:</b>
● <b>Successful Condition:</b> The new sale offer is successfully added to the system.
● <b>Failure Condition:</b> The admin select submit without write the sale information.</td></tr>
</table>

Sequence Diagram:

Sequence Share feedback



Sequence set book offer :

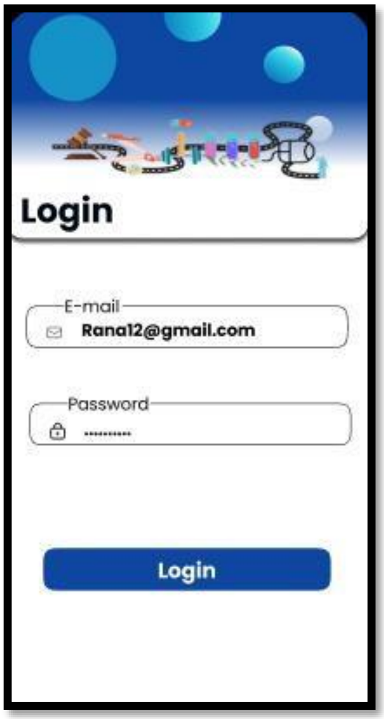

Student
M:bookforsale
setbookOffer()
fillsaleinfo()
fillbookinfo(bookTitle,booktype,price,bookVersion,bookdetalis)
uploadphotoBook()
addCardAcountinfo(accountNumber)
certificated(stateofadd)

System Architecture:

User Interface:

- Login UI

A member's email address and password are required to log in to the system.



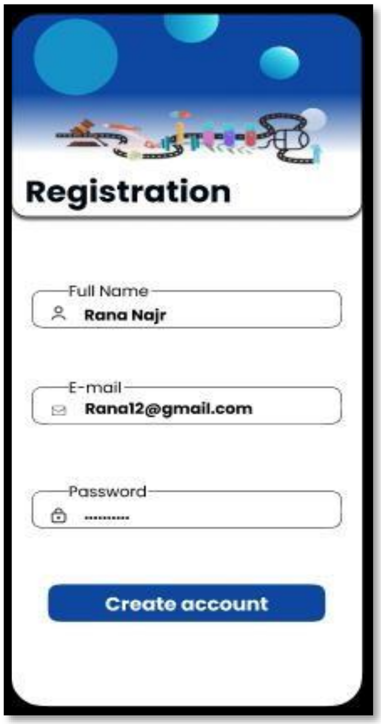
Login

E-mail
Rana12@gmail.com

Password
....

- Register UI

Visitors must fill up their information and hit "create Account" in order to register with the system.



Registration

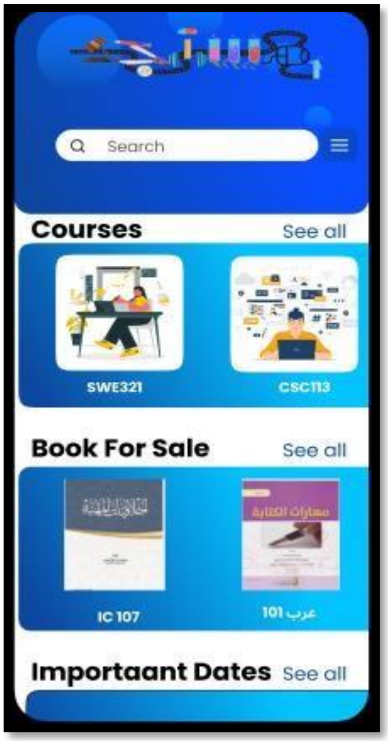
Full Name
Rana Najr

E-mail
Rana12@gmail.com

Password
....

- Search UI

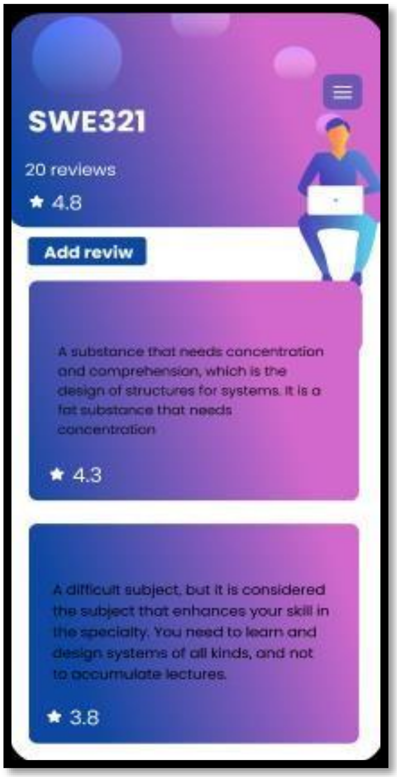
The user may utilize the search box to look for a point of interest once he signs in as a member or decides to create an account. All important points with the specified keywords will show up when they conduct a search for important point .



- view Reviews UI

Through the courses screen, you may access the reviews interface. It will include both the member reviews and the summary review produced by our system.

It's vital to remember that only members are permitted to evaluate and comment on reviews made by other members.



A substance that needs concentration and comprehension, which is the design of structures for systems. It is a fat substance that needs concentration

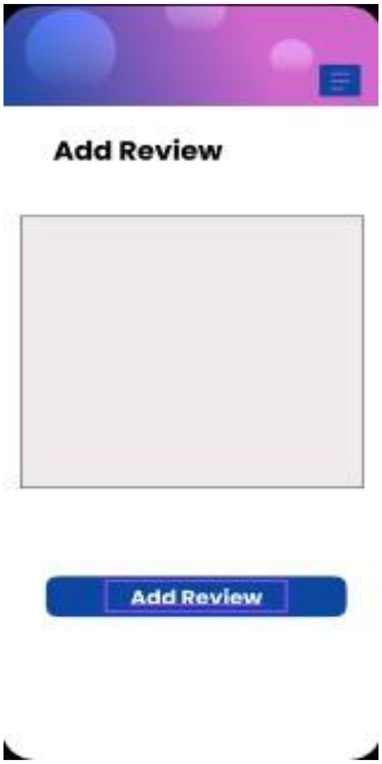
★ 4.3

A difficult subject. but it is considered the subject that enhances your skill in the specialty. You need to learn and design systems of all kinds, and not to accumulate lectures.

★ 3.8

- ## Add Review UI

Only members of our system are able to use the add reviews interface; they may add their review as a description.



Non-Functional Properties:

We will adopt MVC distributed as 3-tier Architecture styles because the MSTAGED System needsto ensure some key features such as:

1. Performance
   Several users will be using the MSTAGED System synchronously so the system shall accommodate 3000 users at the same time.

2. Scalability

We should provide our system with a high level of Scalability. Our system needs to deal with growing amount of users using the system at the same time in the future. Byusing MVC distributed as 3-tier Architecture style will improve Scalability in the system.

3. Reliability
   Because our system will be used through a large number of students it shall be able to recover from failure in less than 2 minutes, so the students don't have to wait for a long period of time.

4. Usability
   Because there is a large number of students will use the system with different knowledge and various backgrounds we should provide a high level of usability for the system. Thus, the users shall learn how to use the MSTAGED System in less than 20 minutes

5. Flexibility
   Our system will have different functionalities that will be discovered in the future so we need to make changes to the system and will take no longer then 1 hour to add new functionality to the system. By using MVC Architecture our changes do not affect the whole system and making the system simple to modify and suitable to add new functionalities.

In conclusion , the system will be running on the IOS platform. Also, our MSTAGED system will be available in Arabic and English languages so we can provide services to a large number of users and increase user’s satisfaction.

Quality Assurance:

Reviews:

As of reviews, we will conduct 2 types of reviews:

Walkthrough :

is an informal review performed by peers. It
need no prior preparation.
They review that work and give their comments in order to
enhance the job and find defects .

Inspection :

is the most formal review type • In an inspection a piece of document is given to group of inspectors in advance with the specific intent of finding errors
in it.

Tests:

As of tests, we will conduct 2 types of tests:

1. Validation :

It is a process of checking that a software achieved its intended goal without any defects. Which gives answers to the rising question: Are we building the right product?
The system will be validated through a validation dynamic test called user acceptance test which is also known as beta or end-user testing, is defined as testing the software by the user or client to determine whether it can be accepted or not.

2. Verification

For each development phase, to ensure that our software met its
specified requirements and fulfilled its intended purpose using software reviews ,inspection and walkthrough, it's do during the development phase also verification process at each stage reviews/walkthrough activities, and finally inspection of the code before its execution. The verification process will demonstrate to us the completeness, correctness, and consistency of the software at each stage.

Acceptance Criteria :

1.Procedures :The user will be able to using system by downloading from apple store .

2.Testing :
1.2- after implementing the program with the help of specialists to test the system and ensure that it complies with the requirements

2 .2- test before the Release of the system is conducted by the end users of the software through their environment. The benefit is to take the users' perspective about system before Releasing the system

3.Training : Provide a video explaining the features of the system that does not exceed five minutes

4.Documentation: It simply placed more value on working software than on comprehensive documentation because of the dynamic nature of software agile developments

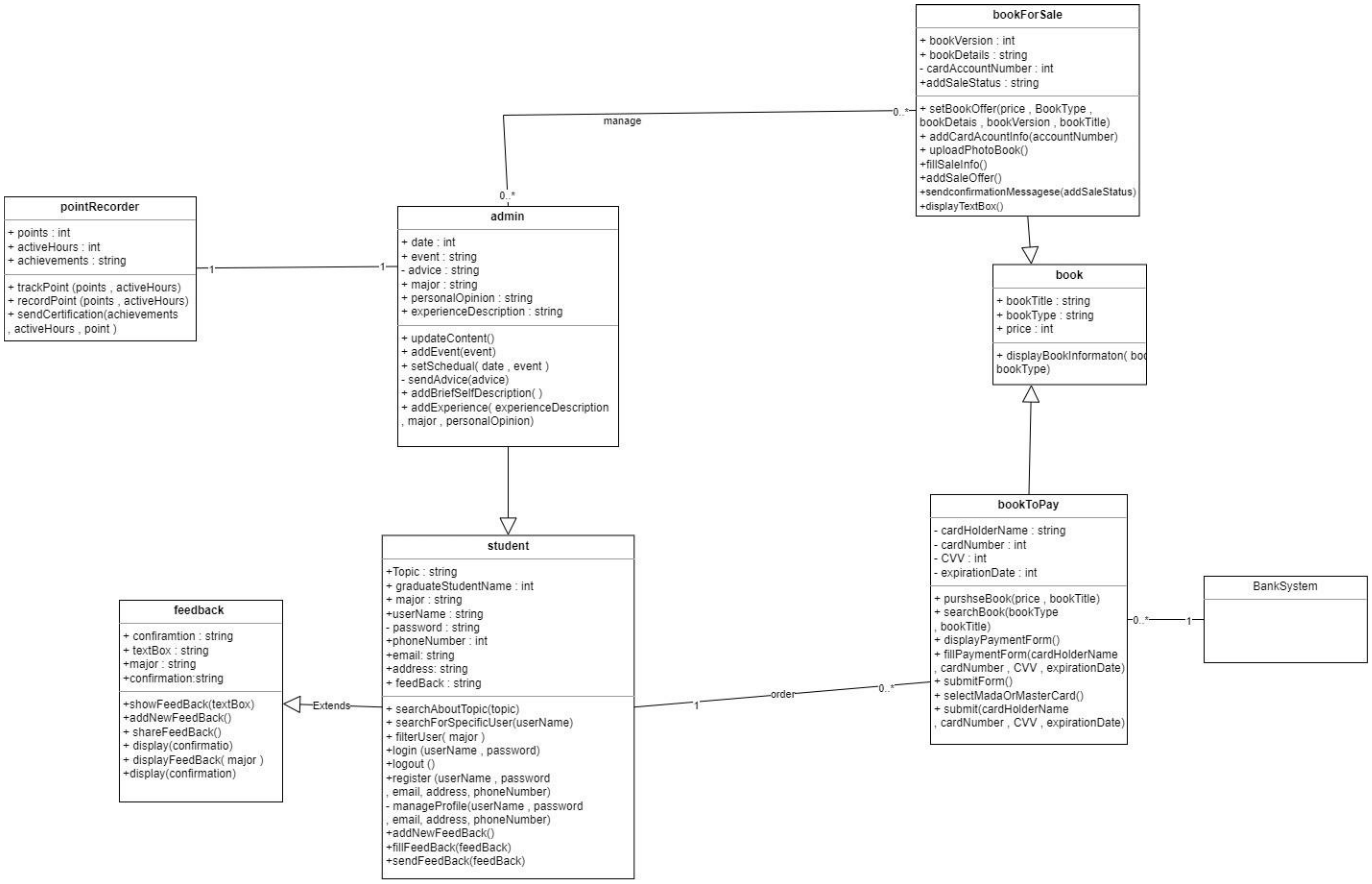
Future Considerations:

Future improvements to our system are unavoidable given the vast domain and wide range of people it serves, and we are fully aware of this. ability for users to have a following and followers that they can interact with on any social networking app is one of the features we would want to introduce. This would necessitate the installation of feeds where users can check their followers' likes.

We also want to add another feature, which is to allow private lessons or YouTube links that explain different topics related to a subject. And we want to add a feature, which is a box that collects the most important activities, hackathons, and events, inside and outside the university, that will increase the knowledge base and gain experiences and talents that add to him in his specialization.

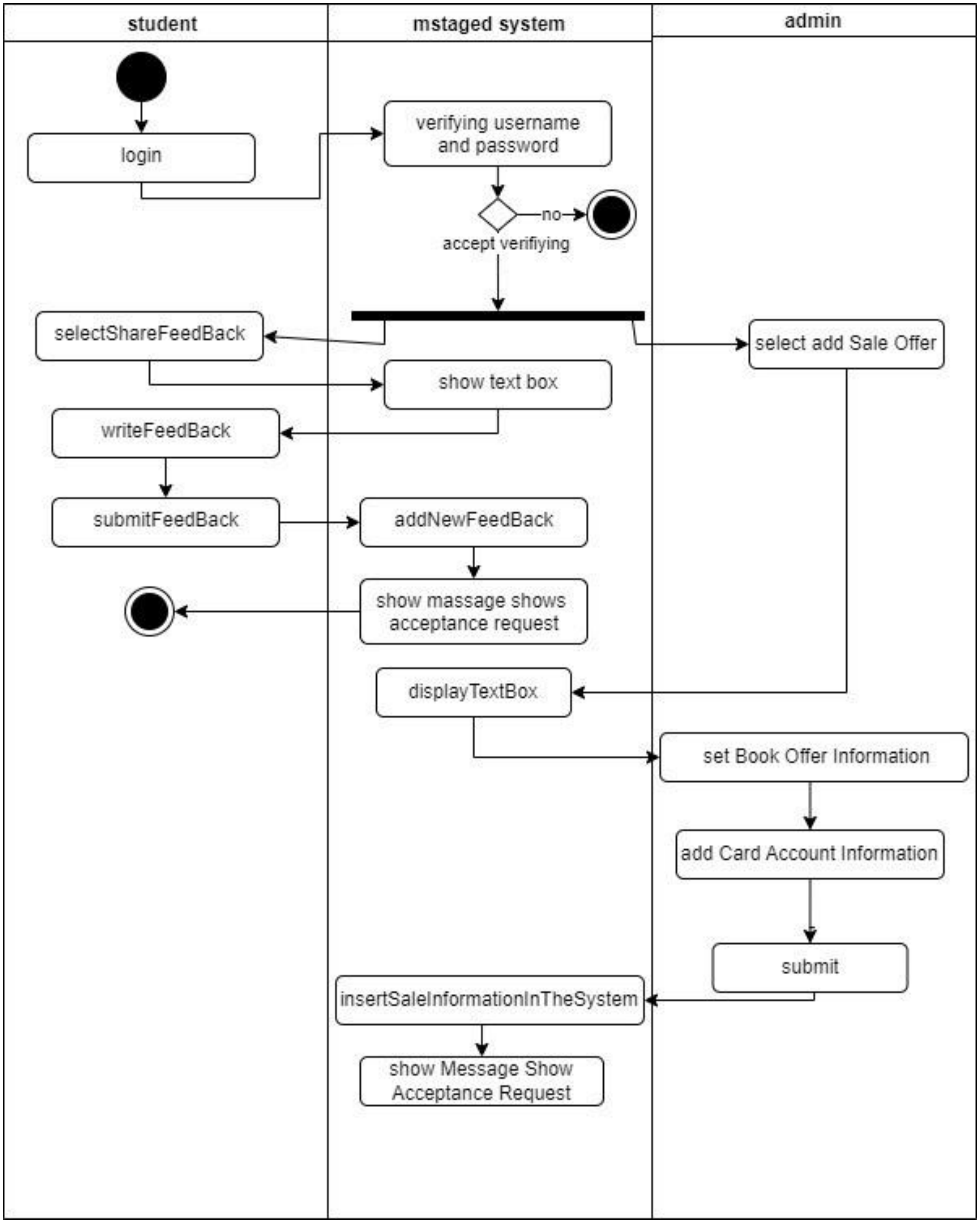
To make sure that users are getting the most out of our system, we want to make any required adjustments that will aid members in accessing and utilizing it. These changes will also increase usability and convenience.

Class Diagram:

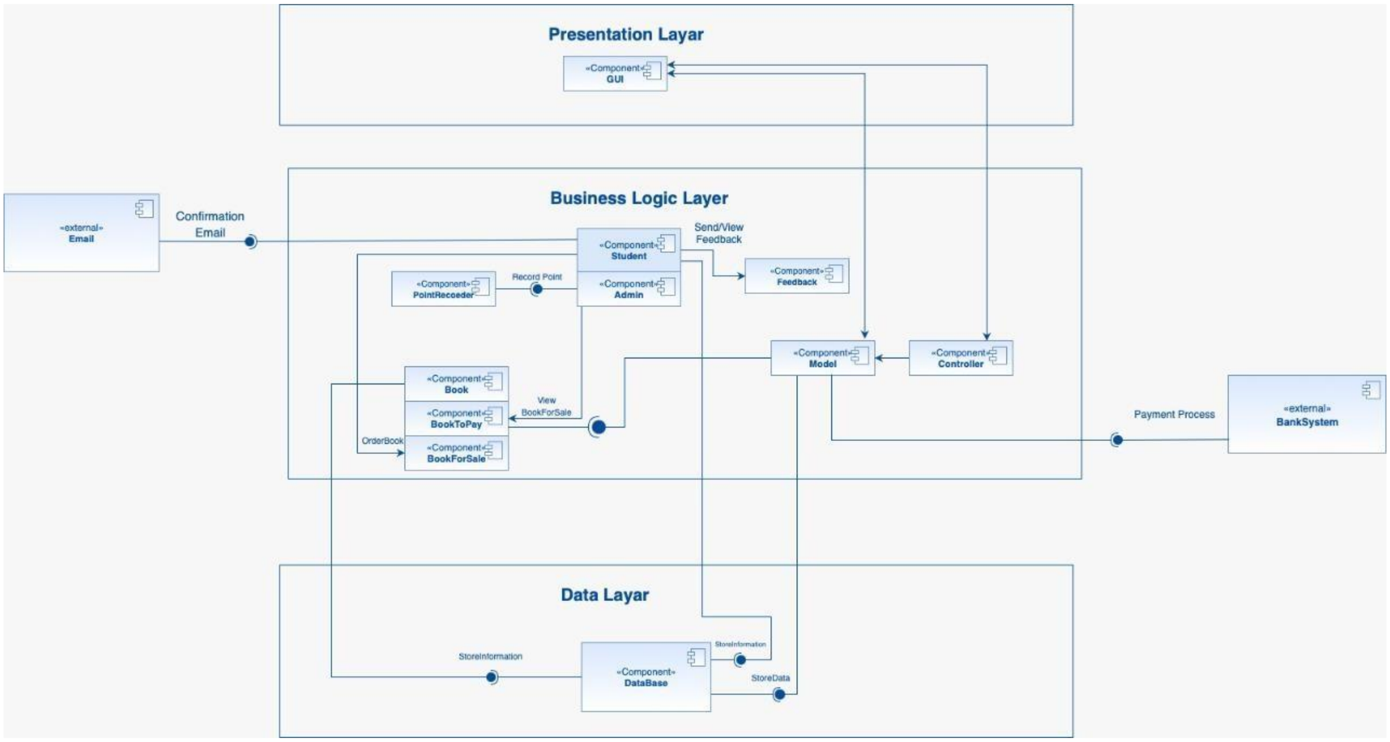

bookForSale
+ bookVersion : int
+ bookDetails : string
- cardAccountNumber : int
+addSaleStatus : string
+ setBookOffer(price , BookType , bookDetais , bookVersion , bookTitle)
+ addCardAcountInfo(accountNumber)
+ uploadPhotoBook()
+fillSaleInfo()
+addSaleOffer()
+sendconfirmationMessagese(addSaleStatus)
+displayTextBox()
manage
0..*
0..*
pointRecorder
+ points : int
+ activeHours : int
+ achievements : string
+ trackPoint (points , activeHours)
+ recordPoint (points , activeHours)
+ sendCertification(achievements , activeHours , point)
1
1
admin
+ date : int
+ event : string
- advice : string
+ major : string
+ personalOpinion : string
+ experienceDescription : string
+ updateContent()
+ addEvent(event)
+ setSchedual(date , event)
- sendAdvice(advice)
+ addBriefSelfDescription( )
+ addExperience(experienceDescription , major , personalOpinion)
book
+ bookTitle : string
+ bookType : string
+ price : int
+ displayBookInformaton(boo
bookType)
bookToPay
- cardHolderName : string
- cardNumber : int
- CVV : int
- expirationDate : int
+ purshseBook(price , bookTitle)
+ searchBook(bookType , bookTitle)
+ displayPaymentForm()
+ fillPaymentForm(cardHolderName , cardNumber , CVV , expirationDate)
+ submitForm()
+ selectMadaOrMasterCard()
+ submit(cardHolderName , cardNumber , CVV , expirationDate)
0..*
1
BankSystem
student
+Topic : string
+ graduateStudentName : int
+ major : string
+userName : string
- password : string
+phoneNumber : int
+email: string
+address: string
+ feedBack : string
+ searchAboutTopic(topic)
+ searchForSpecificUser(userName)
+ filterUser(major)
+login (userName , password)
+logout ()
+register (userName , password , email, address, phoneNumber)
- manageProfile(userName , password , email, address, phoneNumber)
+addNewFeedBack()
+fillFeedBack(feedBack)
+sendFeedBack(feedBack)
1
order
0..*
feedback
+ confiramtion : string
+ textBox : string
+major : string
+confirmation:string
+showFeedBack(textBox)
+addNewFeedBack()
+ shareFeedBack()
+ display(confirmatio)
+ displayFeedBack(major)
+display(confirmation)
Extends

Detailed Design:

Activity Diagram:


student
mstaged system
admin
login
verifying username and password
no
accept verifiying
selectShareFeedBack
show text box
select add Sale Offer
writeFeedBack
submitFeedBack
addNewFeedBack
show massage shows acceptance request
displayTextBox
set Book Offer Information
add Card Account Information
submit
insertSaleInformationInTheSystem
show Message Show Acceptance Request

Component Diagram:



Component Name	Student
Description	A register user that can access the system
Behavior/Functionality	The component is responsible for viewing Feedback and Book component, add username, search, logout, login, register
Interface	Required interface (Student info) form Admin
Connector	Since the component needs to be connected to the internet to get the information, it will be received as a massage.

Component Name	Feedback
Description	A combination of students' opinions about a particular subject in terms of hours and exams
Behavior/Functionality	Insert feedback Show textbook Add new feedback

	Send feedback Display feedback
Interface	provides interface (view feedback) to the student component
Connector	The information will be sent as a massage since the component must be connected to the internet to receive it

Component Name	Bank System
Description	The payment gateway is used to accept debit or credit card purchases from customers.
Behavior/Functionality	. Accept debit or credit card purchases from customers successfully or un successfully
Interface	Provide Process Payment interface to Order controller.
Connector	it is connected with the Book to pay component to take the user input

Component Name	Admin
Description	This component holds manager’s information.
Behavior/Functionality	The main function is to provide the admin the ability to insert book or view, retrieve, update, send message and modify the database.
Interface	: It is connected to GUI, User, and Database.
Connector	It is connected to Point Recorder, student, and Admin profile.

Component Name	controller
Description	It is controllers the mstaged system and connect with GUI component to storeit.
Behavior/Functionality	Initiated the GUI component
Interface	Required interface (mstaged system) to the GUI component
Connector	The information will be sent as massage since the component must be connected to the internet to receive it

Component Name	Point Recorder
Description	He collects the score for the student and calculates his reaction rate
Behavior/Functionality	Track Point, record point, send certification

Interface	Provide interface (point Recorder) to the admin component
Connector	Connected with the admin to collect information and data

Component Name	Book
Description	This component will display the books with the available details about them and include the price
Behavior/Functionality	It will take the information from the Books for Sale component and display it with the addition of the title, type, and price
Interface	Provide interface (Admin) to Admin Access in Book for Sale component
Connector	it relates to the Book to pay and book to sale component to take the user input

Component Name	Book to pay
Description	This component will display the Books that the student wants to pay for
Behavior/Functionality	Purchase book, search book
Interface	Provide interface (order book) to student component
Connector	it relates to the Book to sale and payment component to take the user input

Component Name	Book to sale
Description	This component will allow the student to display his books that he wants to sell
Behavior/Functionality	Add card account info Upload phone Add book for sale
Interface	Provide interface (payment info) to payment component
Connector	it relates to the Book to pay and book to sale component to take the user input

Component Name	Database
Description	The storage component contains all data of the system.

Behavior/Functionality	This component holds all the system data and enables confirmation component to retrieve the information needed from the storage component. Moreover. The admin can access and monitor the system through the database, he\she also can make some changes to the database (cancel appointments).
Interface	Provide interface (Retrieve data) to the mstaged System
Connector	The information will be sent as massage since the component must be connected to the internet to receive it

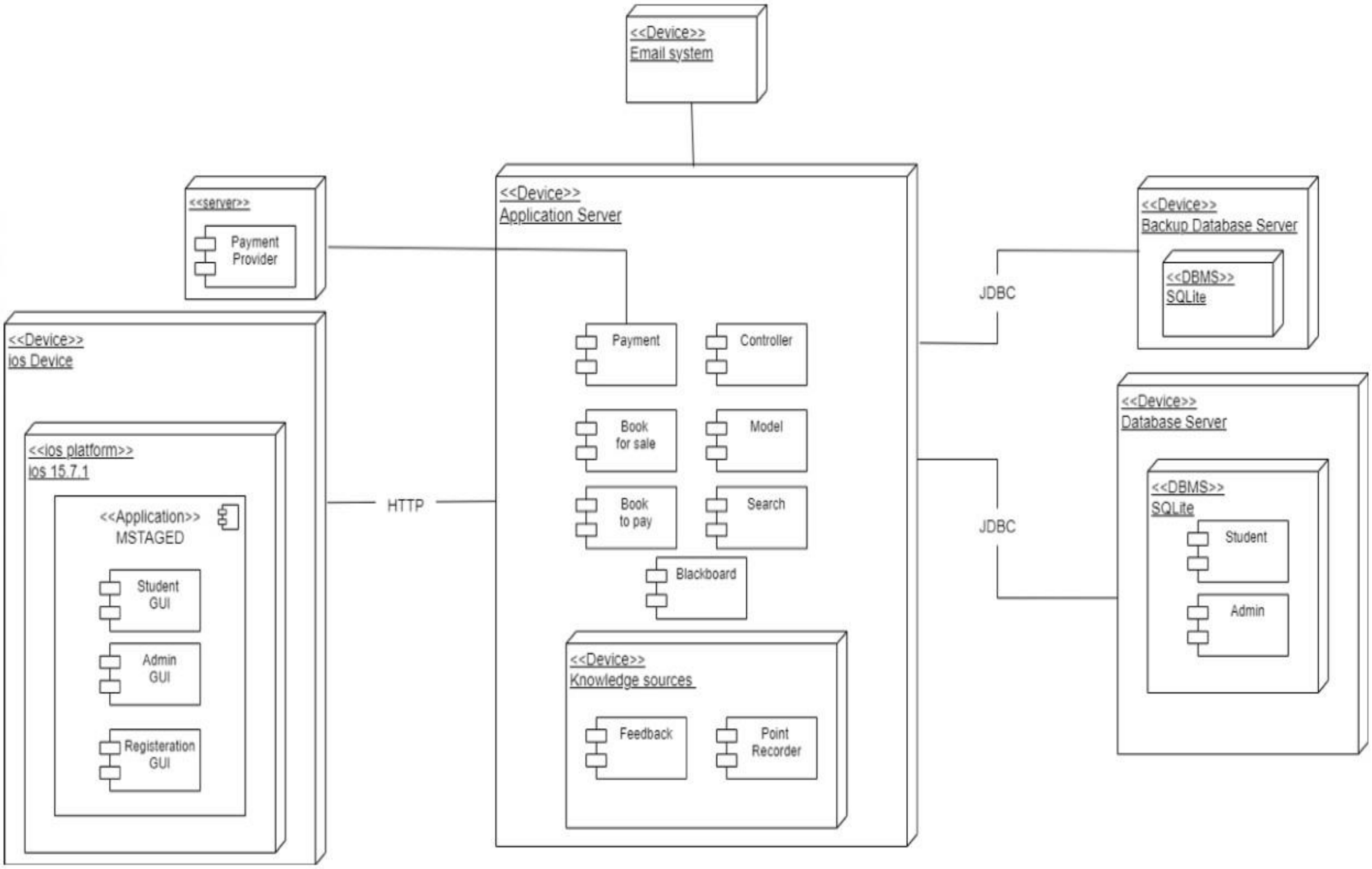
Component Name	Email
Description	The email components are responsible for sending email to student and admin.
Behavior/Functionality	Retrieve Student and admin emails 1-Send confirmation email to student and admin after successful purchase 2-Notify Admin and student of new offers and Important dates.
Interface	1. Requiers Offer Info interface from Adman component. 2. Provides Notification Email interface to student component
Connector	It is connected to admin and student components.

Component Name	GUI
Description	Provides the graphical user interface that the user can connect to controllers through.
Behavior/Functionality	Retrieve Student and admin emails 1-Send confirmation email to student and admin after successful purchase 2-Notify Admin and student of new offers and Important dates.
Interface	Provides user Input interface to mstaged system
Connector	it is connected to mstaged system

Component Name	Model

Description	Responsible for communication with database and system controller.
Behavior/Functionality	It directs the business logic and the functionality of the application
Interface	Requires from the database (provide data) interface
Connector	As for the database it will be a protocol-based connecter

Deployment Diagram:



Refrences :

Hamilton, T. (2022) Agile methodology: What is agile model in software testing?, Guru99. Available at: https://www.guru99.com/agile-scrum-extreme-testing.html (Accessed: October 31, 2022).

Diagrams.net - free flowchart maker and diagrams online (no date) Flowchart Maker &amp; Online Diagram Software. Available at: https://app.diagrams.net/ (Accessed: October 29, 2022).

“Free Online Flowchart Maker,” Figma. [Online]. Available: https://www.figma.com/templates/flowchart-maker/. [Accessed: 01-Nov-2022].

“Diagrams.net - free flowchart maker and diagrams online,” Flowchart Maker &amp; Online Diagram Software. [Online]. Available: https://app.diagrams.net/. [Accessed: 01-Nov-2022].