



SECJ2203: Software Engineering

System Documentation (SD)

UTM Ethics Approval Management System (EAMS)

Version 2.0

Date: 31 / 12 / 2024

Faculty of Computing

Prepared by: Hyperrush

Revision Page

a. Overview:

This document represents **Version 2.0** of the system documentation, encompassing five key sections: **Architectural Rationale**, **Architectural Views**, **Data Design**, **User Interface Design** and **Traceability**. The **Architectural Rationale** section explains the chosen architectural style and the reasoning behind its selection. The **Architectural Views** section offers comprehensive details on various perspectives, including the use case view, implementation view, logical view, process view, and deployment view. The **Data Design** section provides a structured data dictionary that describes every entity within the application. The **User Interface Design** section showcases the proposed application's design through wireframes and prototypes. Finally, the **Traceability** section delivers a detailed cross-reference, mapping components and data structures to the system requirements documented in this version.

b. Target Audience:

UTM staff, student and staff UTM REC.

c. Project Team Members:

Member Name	Role	Task	Status
ABDALLA ALI ABDALLA ALI A23CS3022	Moderator	Monitor group progress	Complete
OTHMAN HASSAN OTHMAN ALI A23CS3026	Recorder	Record the results of discussion	Complete
EYAD AIMEN ELSHEIKH KHALIL A23CS3024	Skeptic	Give query and comment on the assignment	Complete
CHEW CHUAN KAI A23CS0062	Accuracy checker & Reporter	Check and compile the assignment	Complete

d. Version Control History

Version	Primary Author(s)	Description of Version	Date Completed
1.0	ABDALLA ALI ABDALLA ALI	Completed Chapter 1 and 2, Section Introduction and Specific Requirements	25/11/2024
2.0	EYAD AIMEN ELSHEIKH KHALIL	Completed Chapter 3, 4, 5, 6 and 7 Section.	31/12/2024

Table of Contents

1	Introduction		4
	1.1	Purpose	4
	1.2	Scope	4
	1.3	Definitions, Acronyms and Abbreviations	6
	1.4	References	6
	1.5	Overview	7
2	Specific Requirements		8
	2.1	User characteristics	8
	2.2	System Features	8
	2.3	Use Case Details	13
	2.3.1	UC001: Use Case <Apply research>	13
		Use Case Specification of UC001	13
		System Sequence Diagram of UC001	13
		Activity Diagram of UC001	14
	2.3.2	UC002: Use Case <Submit application>	15
		Use Case Specification of UC002	15
		System Sequence Diagram of UC002	15
		Activity Diagram of UC002	16
	2.3.3	UC003: Use Case <Make payment>	17
		Use Case Specification of UC003	17
		System Sequence Diagram of UC003	18
		Activity Diagram of UC003	19
	2.3.4	UC004: Use Case <Track progress>	20
		Use Case Specification of UC004	20
		System Sequence Diagram of UC004	21
		Activity Diagram of UC004	22
	2.3.5	UC005: Use Case <Update progress>	23
		Use Case Specification of UC005	23
		System Sequence Diagram of UC005	23
		Activity Diagram of UC005	24

		2.3.6	UC006: Use Case <Review application>	25
			Use Case Specification of UC006	25
			System Sequence Diagram of UC006	25
			Activity Diagram of UC006	26
	2.3.7		UC007: Use Case <Notify applicant>	27
			Use Case Specification of UC007	27
			System Sequence Diagram of UC007	27
			Activity Diagram of UC007	28
	2.3.8		UC008: Use Case <Get help>	29
			Use Case Specification of UC008	29
			System Sequence Diagram of UC008	29
			Activity Diagram of UC008	30
	2.4		Performance and Other Requirements	31
	2.5		Design Constraints	31
3	Architectural Rationale			32
	3.1		Architectural Style and Rationale	32
4	Architectural Views			33
	4.1		Use Case View	34
	4.2		Implementation View	35
	4.3		Logical View	40
	4.4		Process View	45
	4.5		Physical/ Deployment View	46
5	Data Design			50
	5.1		Data Dictionary	51
		5.1.1	Entity: <Applicant>	51
		5.1.2	Entity: <StaffREC>	51
		5.1.2	Entity: <ChatBot>	51
		5.1.2	Entity: <Dashboard>	51
		5.1.2	Entity: <FormManager>	51
		5.1.2	Entity: <FormPage>	51
		5.1.2	Entity: <UpdatesManagers>	51
		5.1.2	Entity: <Research>	52

		5.1.2 Entity: <PaymentController>	52
		5.1.2 Entity: <RECBankGateway>	52
		5.1.2 Entity: <ReviewFormPage>	53
		5.1.2 Entity: <ReviewController>	53
6	User Interface Design		54
	6.1	Screen Images	54
7	Traceability		60

Introduction:

1.1 Purpose:

This System Documentation (SD) serves to outline the requirements and features of the proposed UTM Ethics Approval Management System (UTM EAMS). It focuses on the System Requirements Specification (SRS), which defines what the system will do and how it is expected to perform. The goal of this documentation is to transition the traditional manual processes of UTM REC into an efficient and automated solution that enhances the overall effectiveness, accuracy, and usability of the research management process.

By leveraging detailed use cases and sequence diagrams, this SD provides a clear understanding of system workflows, interactions, and requirements. These tools help identify potential issues and areas for improvement during development, ensuring alignment with stakeholder needs. The SRS also highlights both functional and non-functional requirements, addressing key aspects such as system performance, usability, and scalability.

This documentation is designed to meet the needs of its primary stakeholders, including **UTM students**, **UTM staff**, and **UTM REC staff**, by capturing their expectations and ensuring the system delivers a user-centric, reliable, and high-performing research management solution. It provides a structured foundation for monitoring progress and ensuring the system meets all requirements efficiently.

1.2 Scope:

The software product is UTM Ethics Approval Management System (UTM EAMS) which is designed to streamline and automate the research ethics approval process for UTM REC. This system caters to three key stakeholders: **UTM students**, **UTM staff**, and **UTM REC staff**, ensuring a seamless and efficient experience for managing research applications. The system offers several essential features, including secure user registration, application submission, payment processing, real-time tracking, and communication modules.

Existing users can log in with their UTM credentials, while new users are required to register using their UTM email, matric number, or staff ID. Verification is enforced via email, ensuring only authorized members gain access. Applicants must also upload identification documents and agree to UTM REC's ethical guidelines, emphasizing the importance of

compliance with research policies. Once registered, users are assigned roles with specific access permissions based on their identities.

The system allows applicants to submit research applications via an online portal, where fields are auto-validated to minimize errors. Applications are categorized by research type (e.g., clinical, non-clinical, or animal research) and routed to the appropriate subcommittees for review. Secure payment functionality is integrated, enabling applicants to pay fees through third-party gateways such as e-wallets, online banking, or debit cards. Payment statuses are updated instantly, and receipts are emailed to applicants while maintaining a comprehensive transaction log.

A dashboard provides applicants with real-time updates, including application progress, estimated processing times, and pending actions. Automated email notifications keep users informed at each stage. For high-risk applications, the system notifies applicants to schedule pitching sessions, which are managed collaboratively between applicants and staff.

For UTM REC staff, the system automates routine tasks such as email communication and payment verification, freeing time for more critical responsibilities. Staff can track application progress, facilitate high-risk reviews, and provide live chat support for applicants. The system ensures that all updates are synchronized across dashboards for both applicants and staff, improving transparency and efficiency.

In addition to these features, the system incorporates a live chat and chatbot module to assist users with common inquiries or escalate complex issues to staff. Our goal is to develop a research management system that enhances operational efficiency, ensures data accuracy, and elevates user satisfaction. By transitioning from manual processes to an automated solution, this system aligns with UTM REC's mission to uphold research integrity and streamline the ethics approval process.

1.3 Definitions, Acronyms and Abbreviation:

Term	Definition
UTM EAMS	The software product being developed to manage the research integrity and streamline the ethics approval process.
SRS	System Requirements Specification - a document that outlines the requirements and objectives of the software product being developed.
SD	System Documentation
UTM REC	UTM Research Ethics Committee
ID	Identity Document

1.4 References:

- [1] Nishadha. (2015, February 17). *Use Case Diagram Relationships Explained with Examples*. Creately Blog. <https://creately.com/blog/diagrams/use-case-diagram-relationships/>
- [2] Visual Paradigm. (2019). *What is Sequence Diagram?* Visual-Paradigm.com. <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/>
- [3] Visual Paradigm. (2019). *What is Activity Diagram?* Visual-Paradigm.com. <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/>
- [4] Domain Modeling: What you need to know before coding. (2021, September 29). Thoughtworks. <https://www.thoughtworks.com/insights/blog/agile-project-management/domain-modeling-what-you-need-to-know-before-coding>
- [5] Altexsoft. (2023, March 28). *Technical Documentation in Software Development: Types and T.* AltexSoft. <https://www.altexsoft.com/blog/technical-documentation-in-software-development-types-best-practices-and-tools/>
- [6] altexsoft. (2022, July 26). *Non-functional Requirements: Examples, Types, How to Approach.* AltexSoft. <https://www.altexsoft.com/blog/non-functional-requirements/>

1.5 Overview:

This System Documentation (SD) outlines the specific requirements gathered from stakeholders for the development of our proposed software. It covers key aspects such as user characteristics, system features, use case details, performance and other requirements as well as design constraints.

For user characteristics, we detail the intended users of the system, including their technical expertise, knowledge levels, and physical abilities. The system features are explained through various diagrams, such as use case diagrams, activity diagrams, sequence diagrams, domain model class diagrams and state machine diagrams, providing a comprehensive understanding of the system's functionality.

Performance and other requirements as well as design constraints are elaborated in detail within this document. The SD is structured to follow the Software Development Life Cycle (SDLC) phases, including requirement analysis, design and testing, with each phase thoroughly documented using diagrams and descriptive explanations.

2. Specific Requirements:

2.1 User characteristics:

In UTM EAMS we have 2 main users of the system, which are: applicants can be a UTM student or UTM staff member and staff REC.

2.1.1 Applicant:

- The applicants must at least be doing a master degree.
- The applicants who will use the software are expected to have basic computer skills, including familiarity with web-based applications.
- The applicants are expected to have experience in email communications.

2.1.2 Staff REC:

- The staff REC are expected to have experience with web-based applications.
- Some of the staff REC are expected to need software training on how to use the new system.
- The staff REC are expected to know how to provide help for the applicants.

2.2 System Features

The UTM EAMS is a web-based application software, this system is accessed via web and internet. The system will allow the applicants to apply, submit and track their research in one platform. The system will also provide one platform for staff REC to process, update all the research applications and notify the applicant if there are any changes required.

The system features are illustrated in Figure 2.1 below. The detailed description of each module and functions is tabulated in Table 2.1.

Figure 2.1: Use Case Diagram for UTM EAMS

Use case Diagram

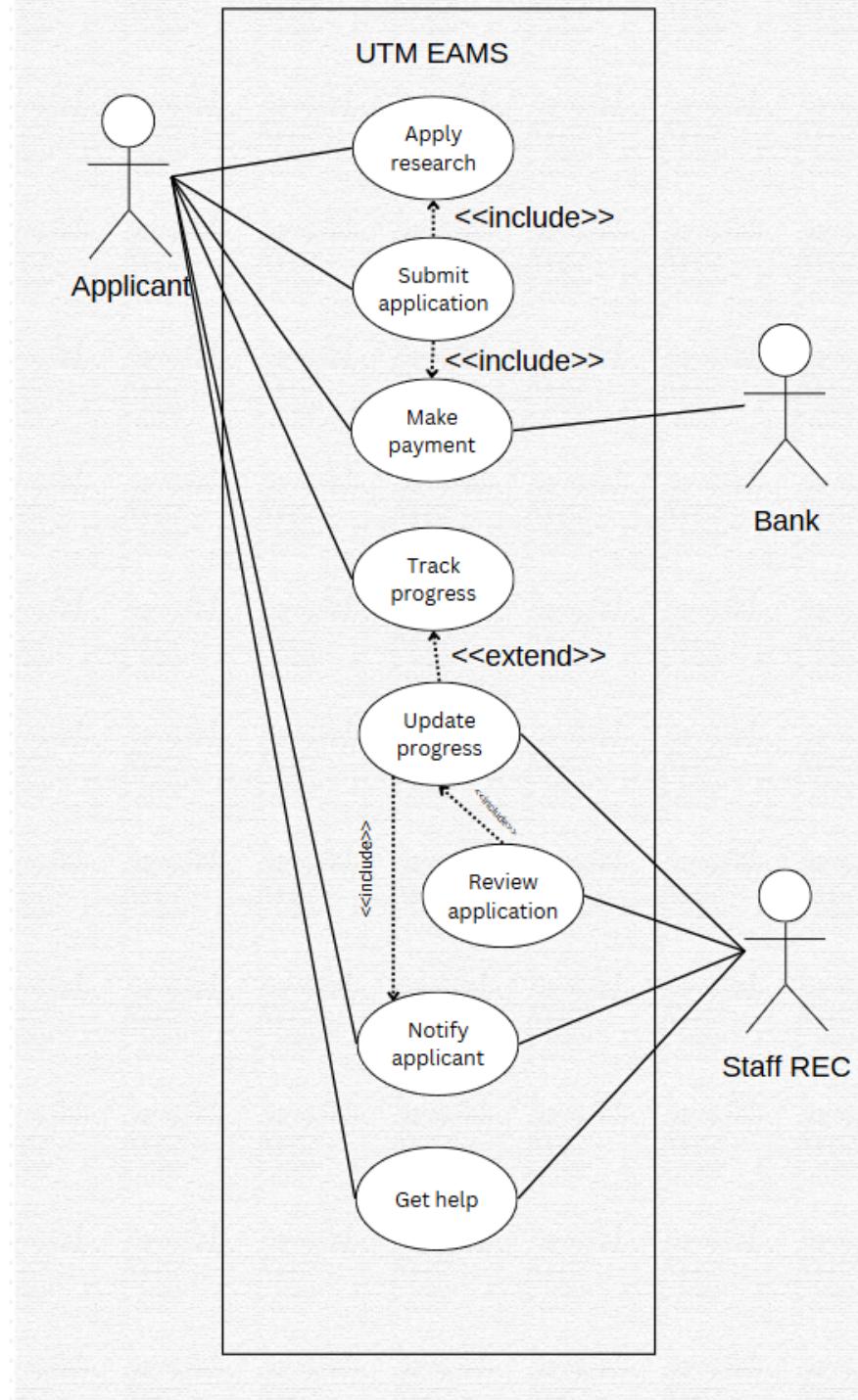


Table 2.1: Description of Module and Functions for UTM EAMS

Module	Function	Description
	UC001 - Apply research	This use case allows the applicants to fill in the research application form
	UC002 - Submit application	This use case allows the applicants to submit the research application form.
	UC003 - Make payment	This use case allows the applicants to pay for the research application form via third party gateway
	UC004 - Track progress	This use case allows the applicants to track their submitted research applications and see all the updates regarding the reviewing process.
	UC005 - Update progress	This use case allows the staff REC to update the status of the research in the reviewing process.
	UC006 - Review application	This use case allows the staff REC to view all the research information and review it.
	UC007 - Notify applicant	This use case allows the staff REC to notify the applicant of the updates regarding the applicant research.
	UC008 - Get help	This use case allows the applicants to get help regarding their issues in the system, they will be assisted either with the system chatbot or one of staff REC members.

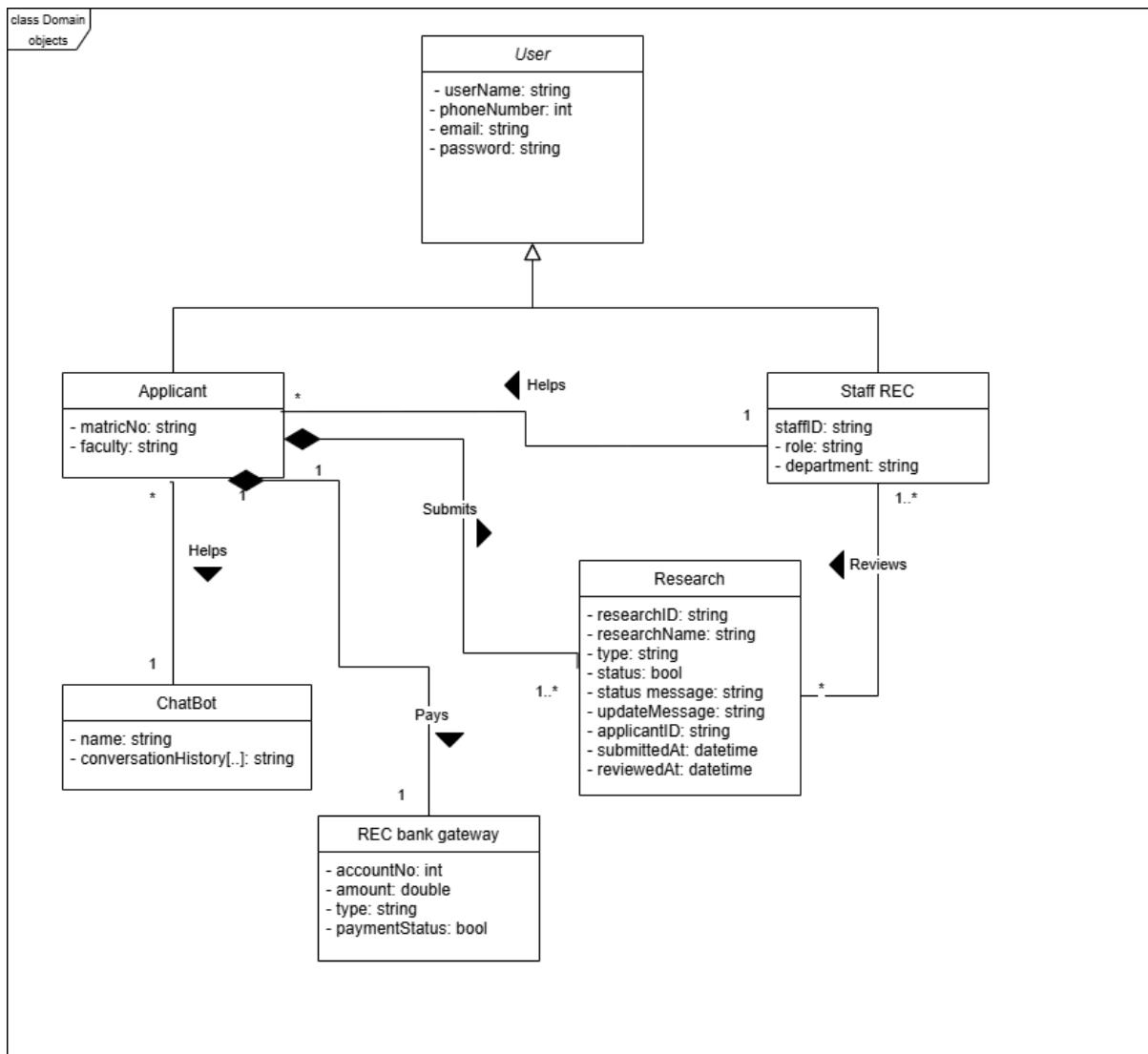


Figure 2.2: Domain Model for UTM EAMS

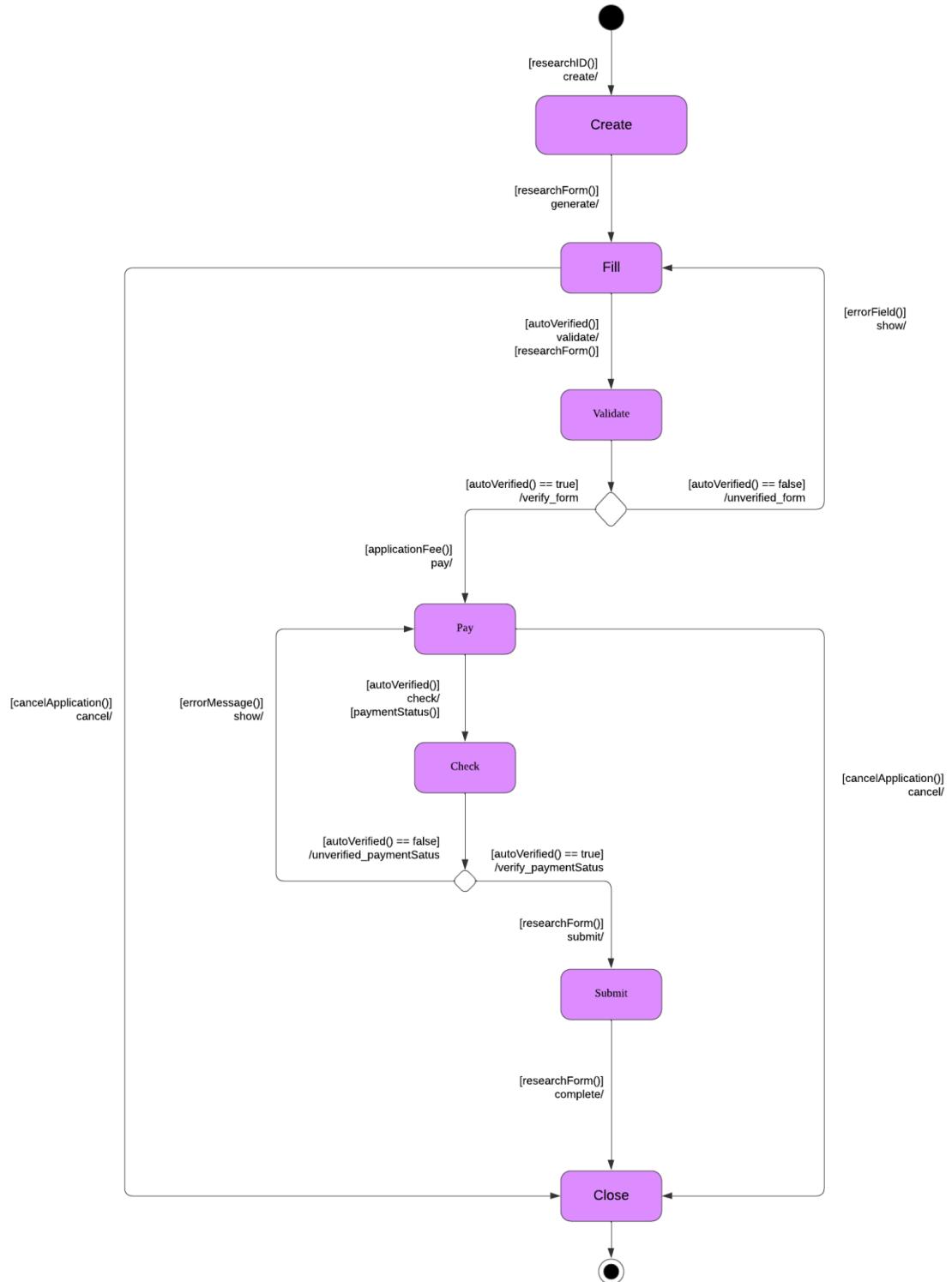


Figure 2.3: State Diagram for Research class

2.3 Use Case Details:

2.3.1 UC001: Use Case Apply research:

Table 2.3.1: Use Case Description for Apply research

Use case: Apply research
ID: UC001
Actors: Applicant (consists of Student and UTM Staff)
Includes: UC002 Submit application
Preconditions: Applicant is already registered in the system
Flow of events: <ol style="list-style-type: none"> 1. The system provides the applicant a sample form to help the applicant fill in the form 2. The applicant fills in the required information. 3. The system checks the input fields. 4. if there was an error in input fields <ol style="list-style-type: none"> 4.1 for each input field that has an error <ol style="list-style-type: none"> 4.1.1 The system displays an error message 4.1.2 The applicant fills in the required information 5. Include (Submit application)
Alternative flow: The applicant cancels the application.
Postconditions: The system returns the applicant to the home page.

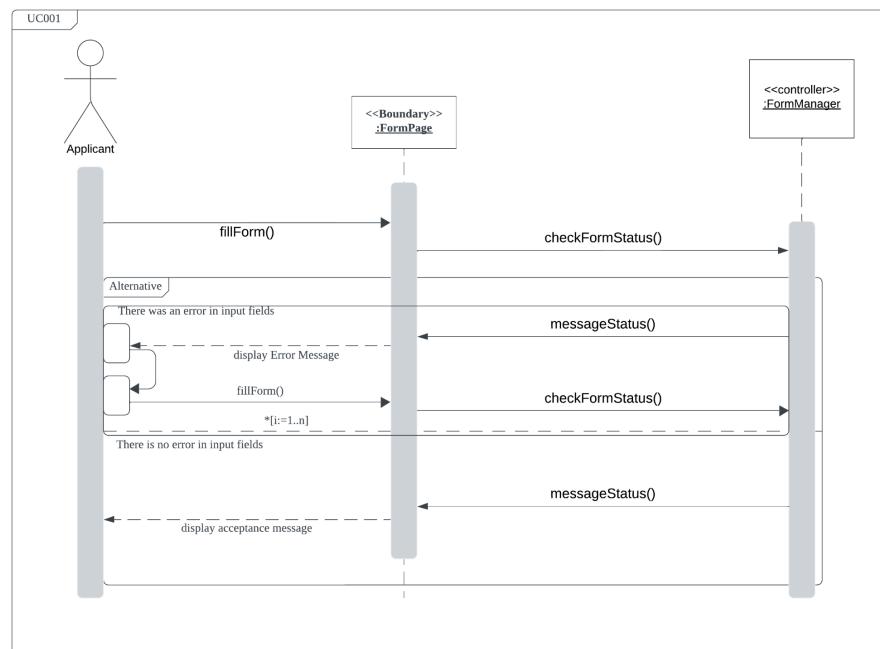


Figure 2.3.1: Sequence Diagram for Apply research

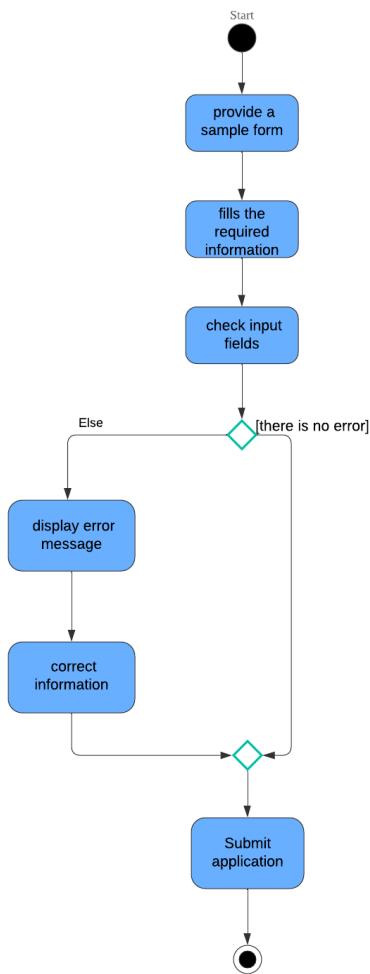


Figure 2.3.2: Activity Diagram for Apply research

2.3.2 UC002: Use Case Submit application

Table 2.3.2: Use Case Description for Submit application

Use case: Submit application
ID: UC002
Actors: Applicant (consists of Student and UTM Staff)
Includes: UC003 Make payment
Preconditions: The application form fields are error free.
Flow of events: <ol style="list-style-type: none"> 1. The applicant clicks the “Submit” button. 2. Include (Make payment). 3. The system categorizes the application based on research clinical, non-clinical and animal research. 4. The application is routed to the respective subcommittee for review. 5. The applicant receives an email notification confirming successful submission.
Postconditions: The applicant’s dashboard is updated to reflect the submission status and next steps.

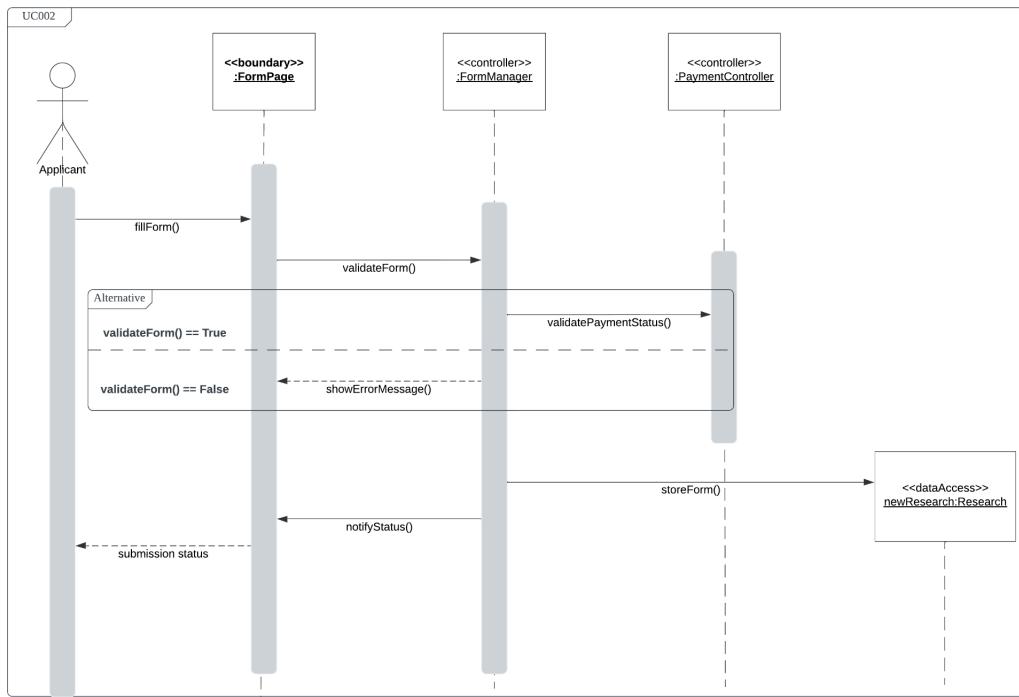


Figure 2.3.3: Sequence Diagram for Submit application

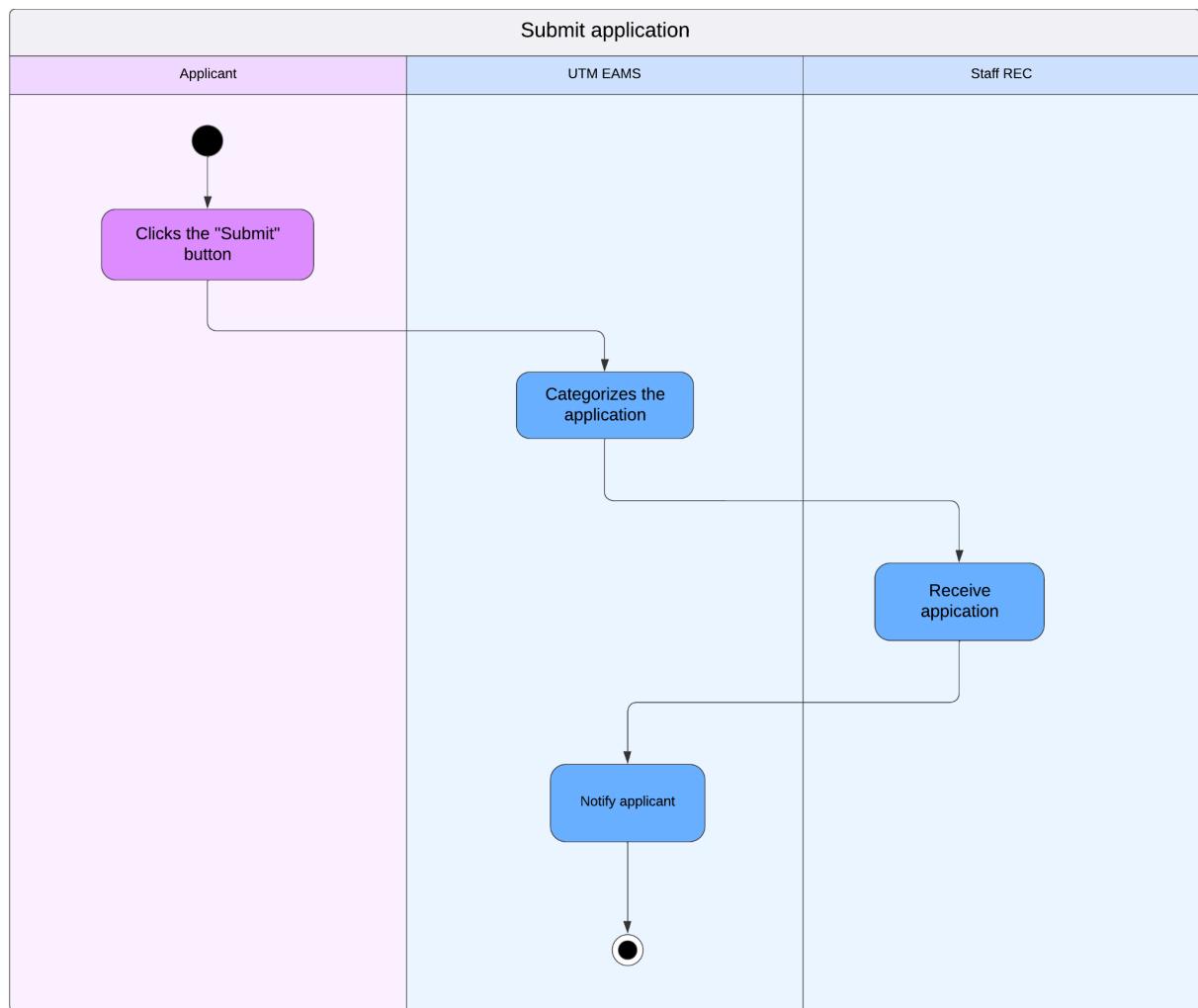


Figure 2.3.4: Activity Diagram for Submit application

2.3.3 UC003: Use Case Make payment

Table 2.3.3: Use Case Description for Make payment

Use case: Make payment
ID: UC003
Actors: Applicant (consists of Student and UTM Staff) Bank
Preconditions: The application form has been completed and validated.
Flow of events: <ol style="list-style-type: none"> 1. The system redirects the applicant to a third-party payment gateway. 2. The applicant chooses a payment method either e-wallet, online banking or debit card. 3. The applicant completes the transaction. 4. If there is a technical issue with the payment gateway. <ol style="list-style-type: none"> 4.1. The system advises the applicant to try again later. 5. If the transaction is fail <ol style="list-style-type: none"> 5.1. The system alerts the applicant with an error message “Transaction Fails”. 6. If the transaction is successful <ol style="list-style-type: none"> 6.1. The payment gateway processes the payment and sends a confirmation to the system. 6.2. The system updates the applicant’s record with the payment status. 6.3. An email receipt is sent to the applicant for confirmation.
Postconditions: <ol style="list-style-type: none"> 1. Payments status is updated as “Completed” in the system. 2. The applicant’s dashboard reflects the updated payment status.
Alternative flow: At any given time the applicant has the ability to cancel the research application

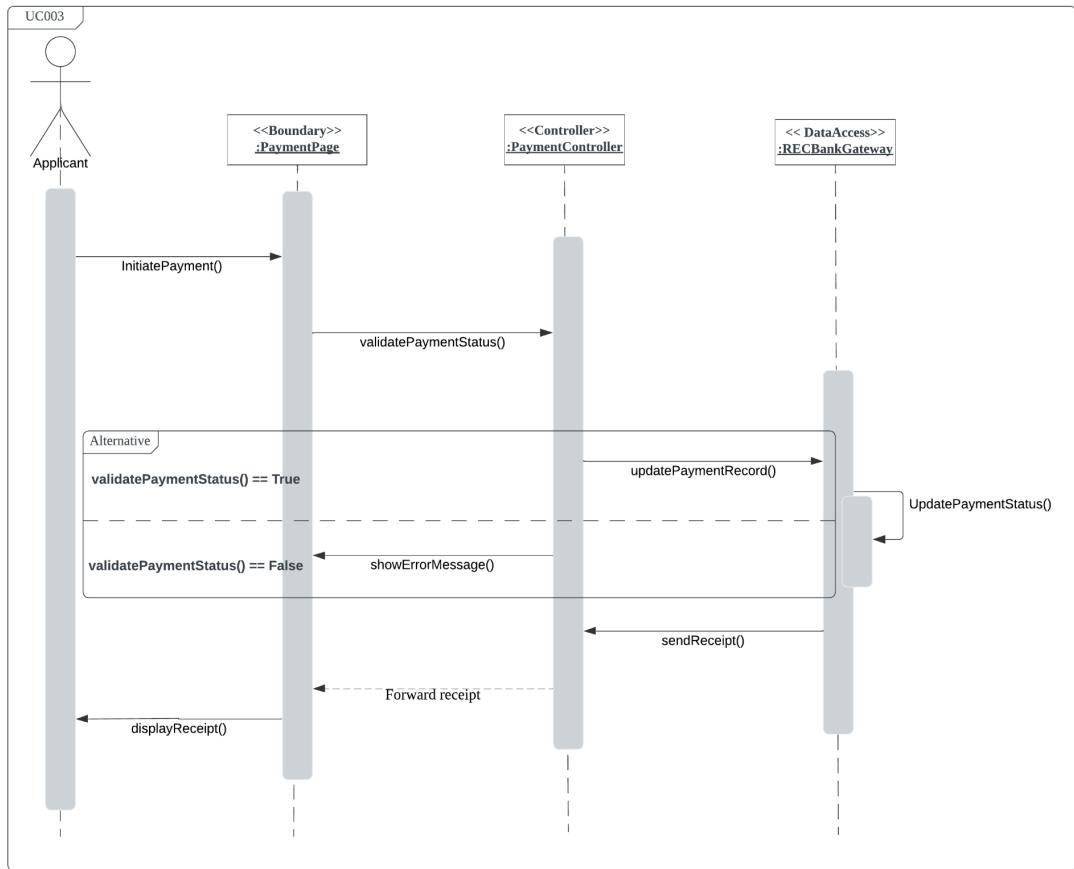


Figure 2.3.5: Sequence Diagram for Make payment

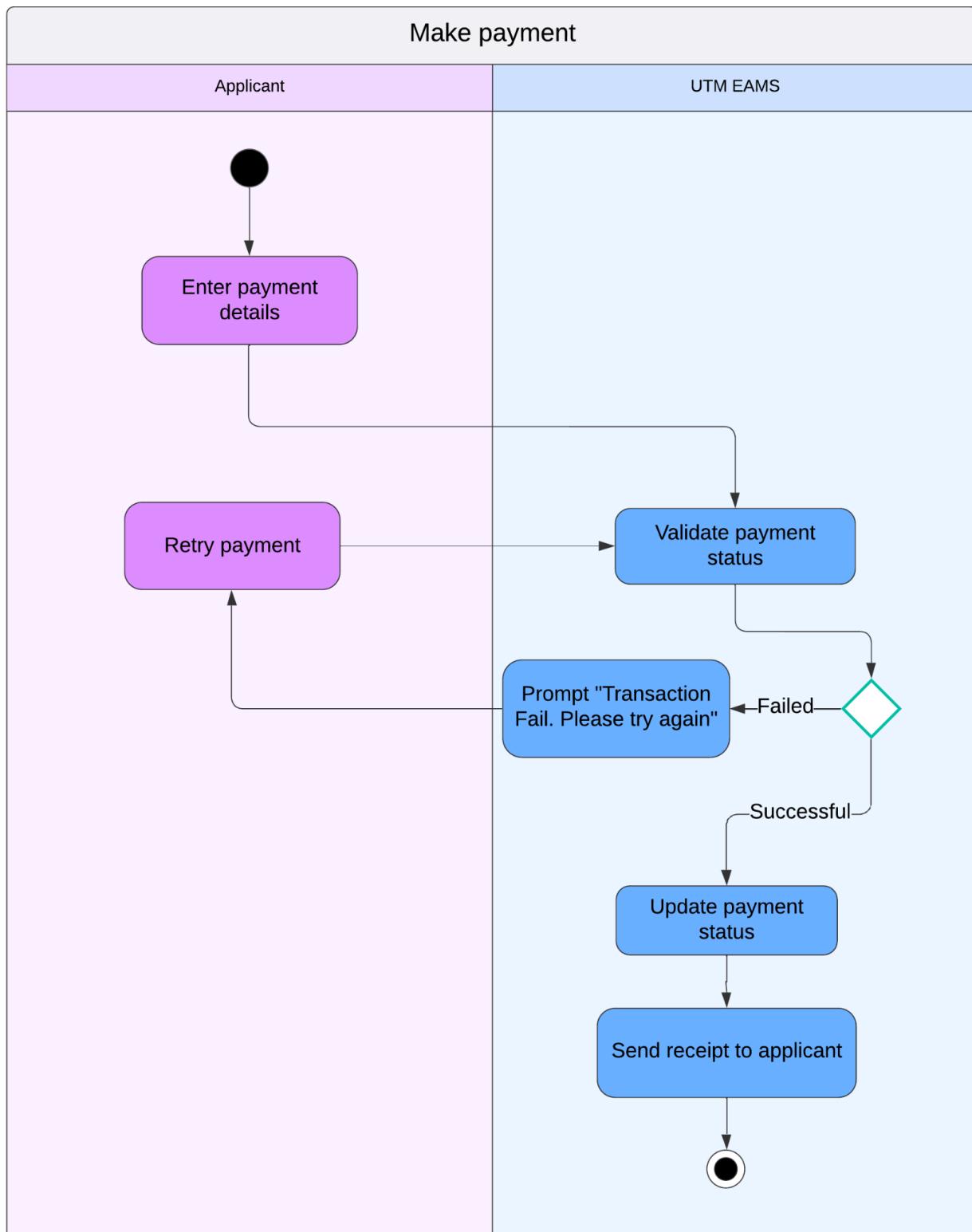


Figure 2.3.6: Activity Diagram for Make payment

2.3.4 UC004: Use Case Track progress:

Table 2.3.4: Use Case Description for Track progress

Use case: Track progress
ID: UC004
Actors: Applicant (consists of Student and UTM Staff)
Extension points: <Updated progress>
Preconditions: 1. The applicant must submit the research.
Flow of events: 1. The applicant opens the dashboard. 2. The applicant selects the research that wants to track. 3. The system checks if there were any updates from the UTM REC staff on the research. 4. If <Updated progress>, then 4.1 The system tells the applicant that there was an update and displays an update message. 5. else 5.1 The system displays there are no updates message 6. The system displays estimated processing time and previous updates on the research
Postconditions:

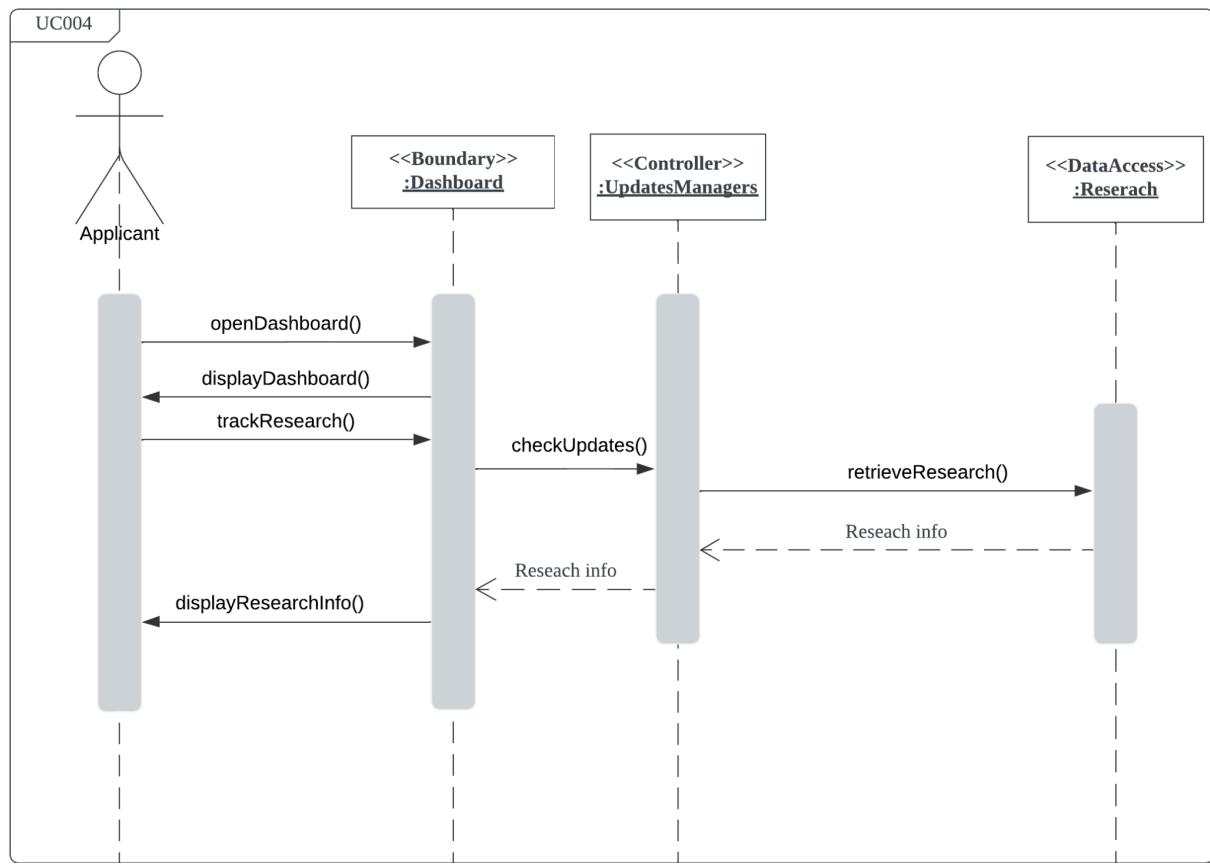


Figure 2.3.7: Sequence Diagram for Track progress

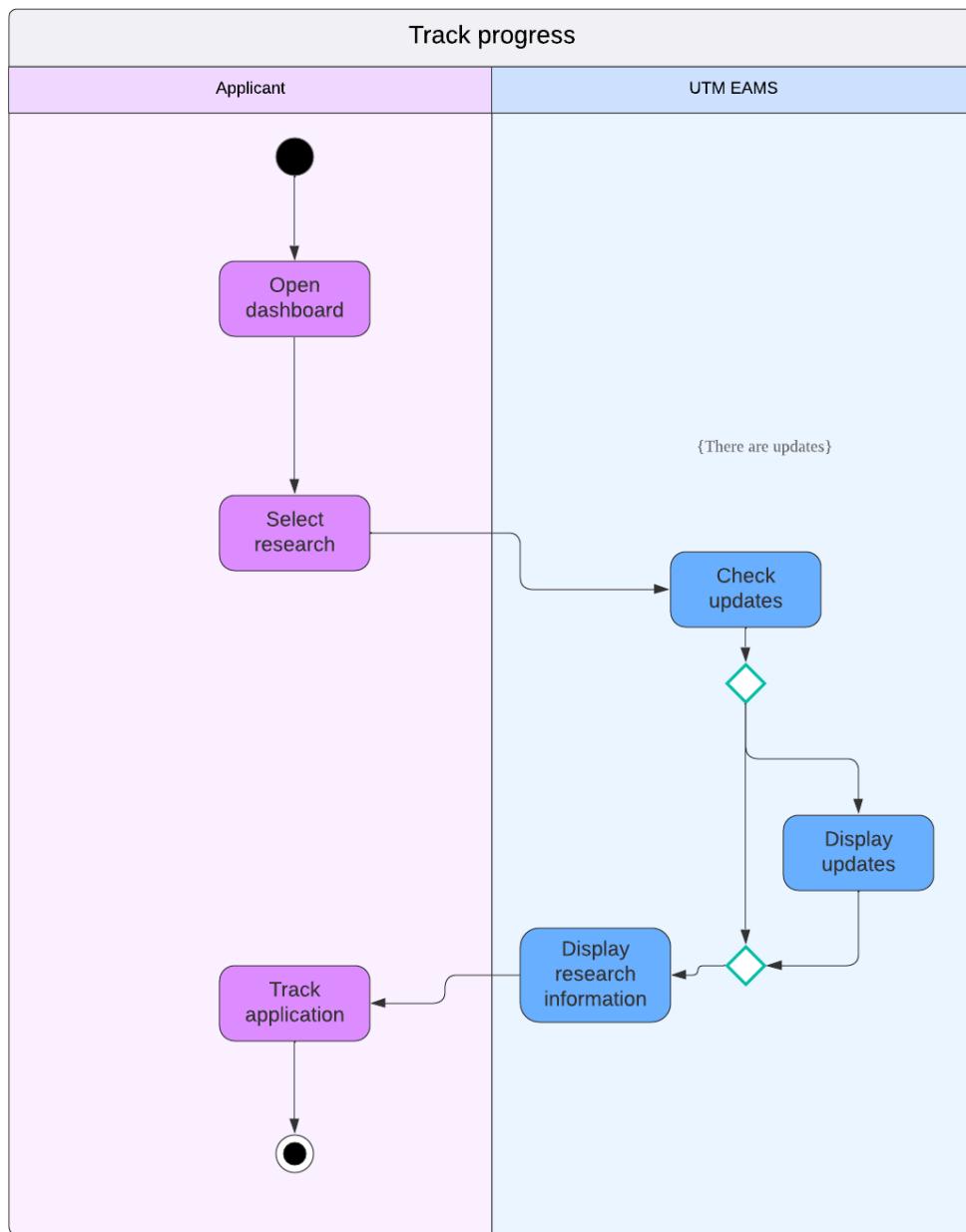


Figure 2.3.8: Activity Diagram for Track progress

2.3.5 UC005: Use Case Update progress:

Table 2.3.5: Use Case Description for Update progress

Use case: Update progress
ID: UC005
Actors: Staff REC
Includes: UC007 Notify applicant
Extends: UC004 Track progress at <Updated progress>
Preconditions: 1. The staff REC must review the submitted research
Flow of events: 1. The Staff REC selects the status flag. 2. The staff REC writes the status update message. 3. Include(Notify applicant). 4. The system displays a message indicating that the update is a success.
Alternative flow: The system fails to submit or save the updated status.
Postconditions: The staff REC has to resubmit the updated progress.

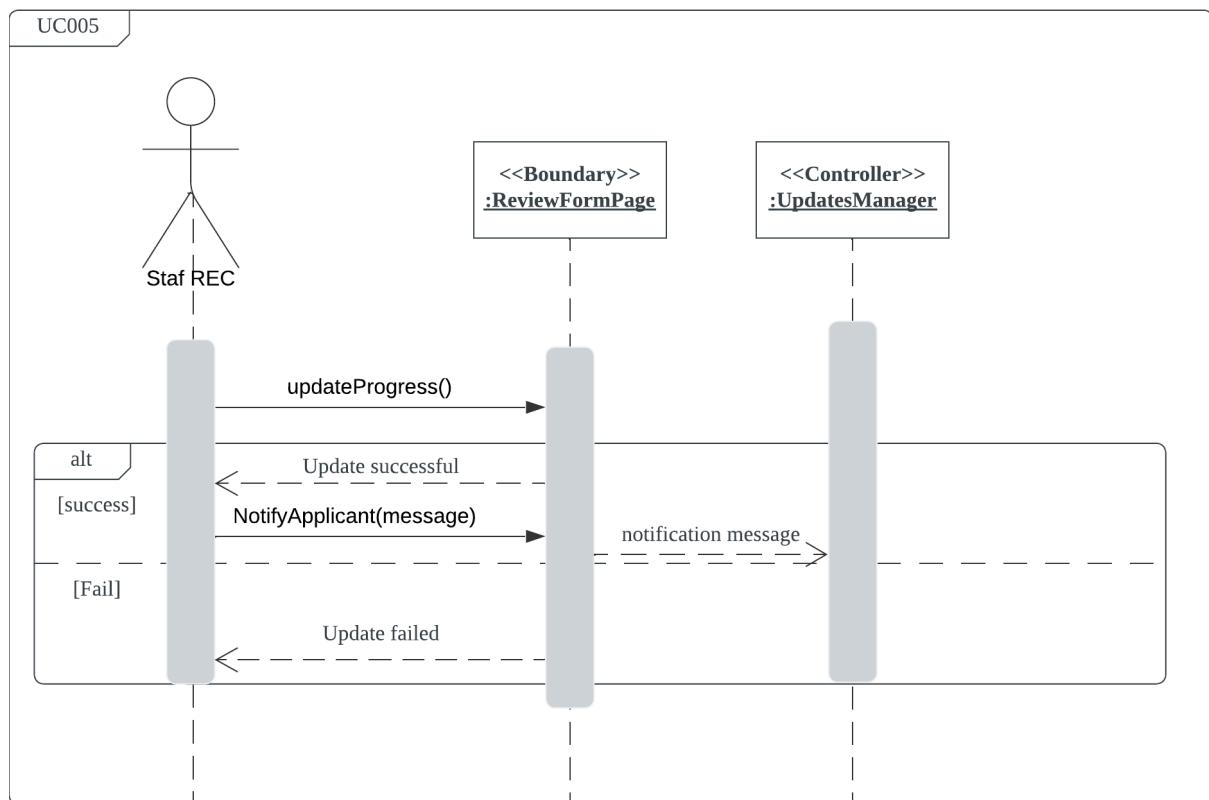


Figure 2.3.9: Sequence Diagram for Update progress

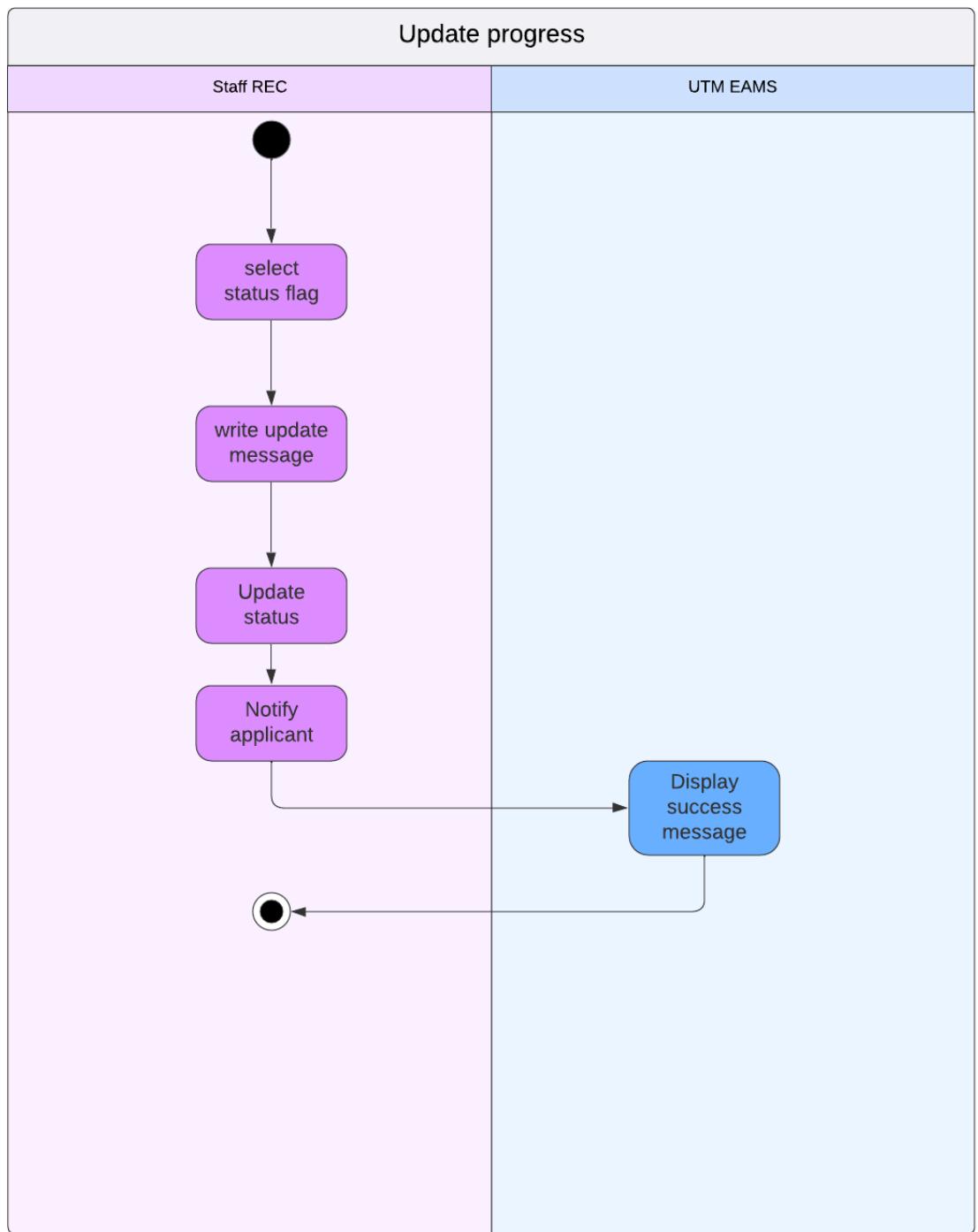


Figure 2.3.10: Activity Diagram for Update progress

2.3.6 UC006: Use Case Review application:

Table 2.3.6: Use Case Description for Review application

Use case: Review application
ID: UC006
Actors: Staff REC
Includes: UC004 Update progress
Preconditions:
Flow of events: <ol style="list-style-type: none"> 1. Staff REC reviews the application. 2. if Staff REC detects high-risk research applications. <ol style="list-style-type: none"> 2.1 The system automatically sends email to applicants to select the available time slot for the review session. 3. Include(Update progress).
Preconditions: 1. The updates should be reflected in the applicant dashboard.

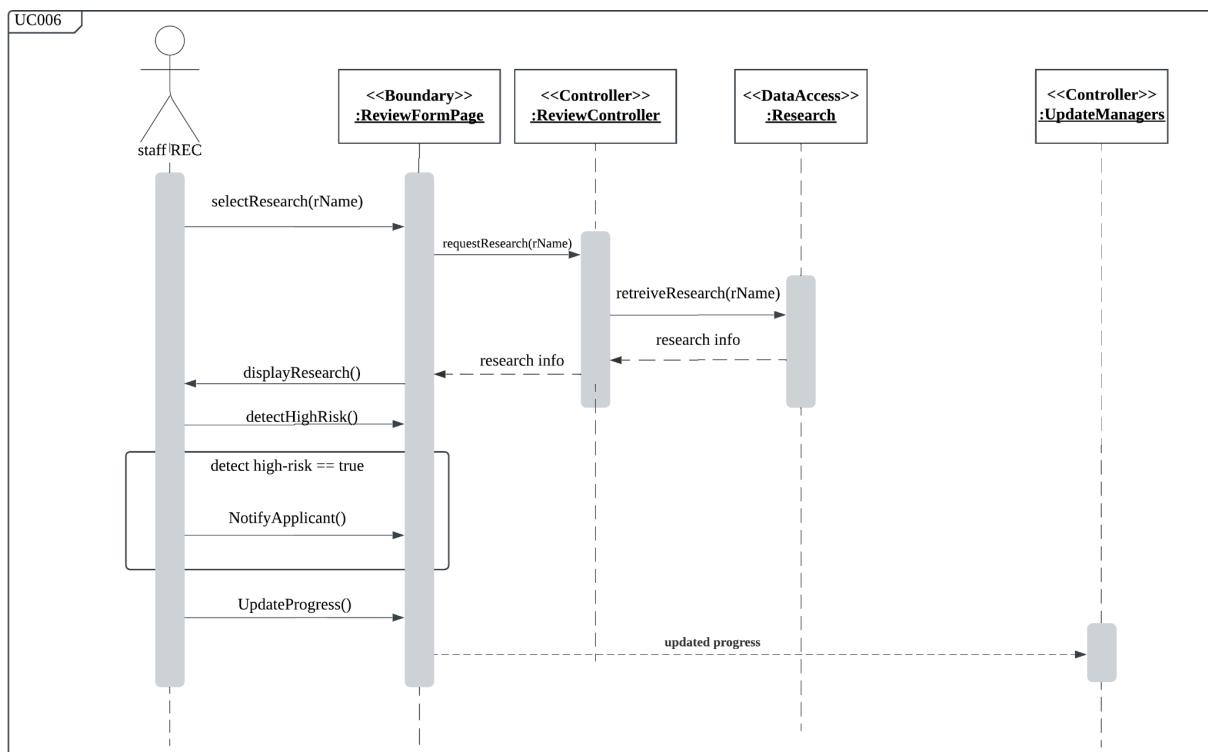


Figure 2.3.11: Sequence Diagram for Review application

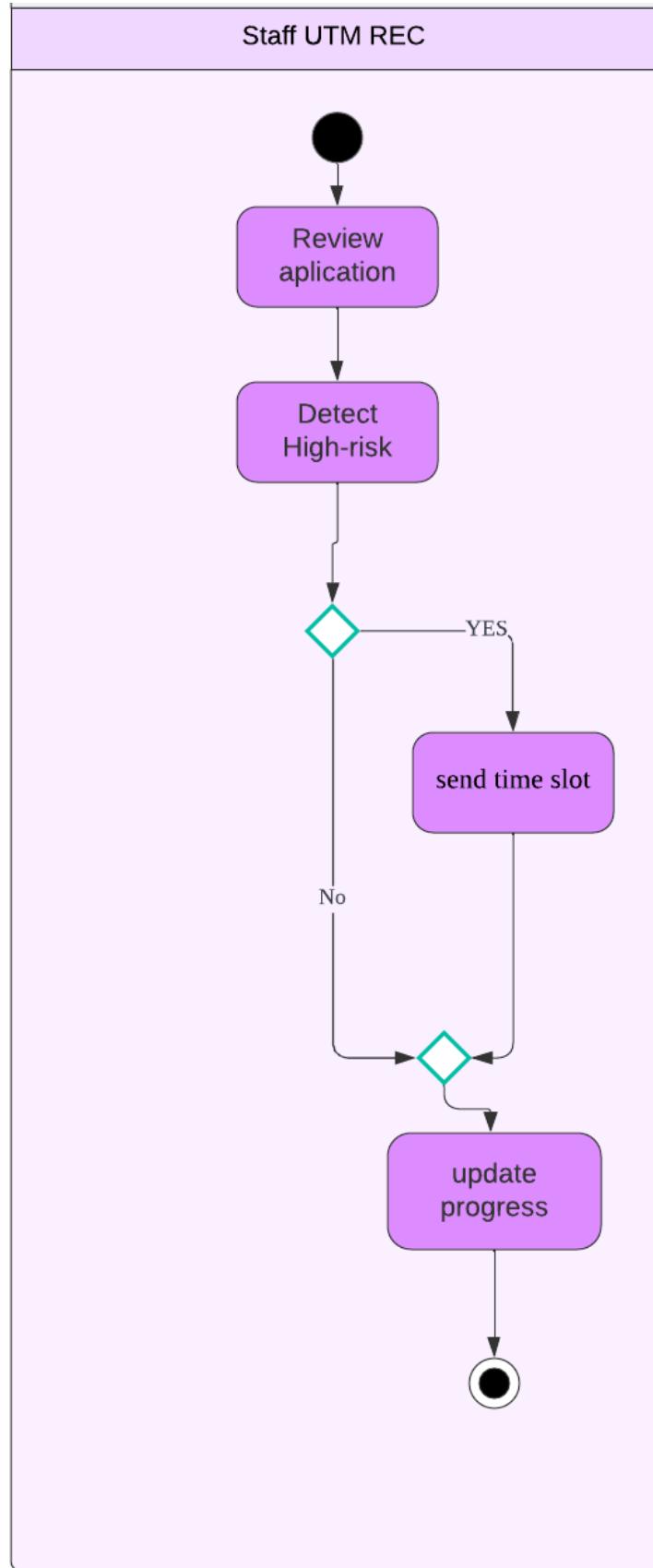


Figure 2.3.12: Activity Diagram for Review application

2.3.7 UC007: Use Case Notify applicants:

Table 2.3.7: Use Case Description for Notify applicants

Use case: Notify applicants
ID: UC007
Actors: Staff REC Applicant
Preconditions: 1. The system progress is updated
Flow of events: 1. staff REC sends notification to the applicant's email regarding the updated progress. 2. applicants receive the email notification indicating that the progress is updated.
Postconditions:

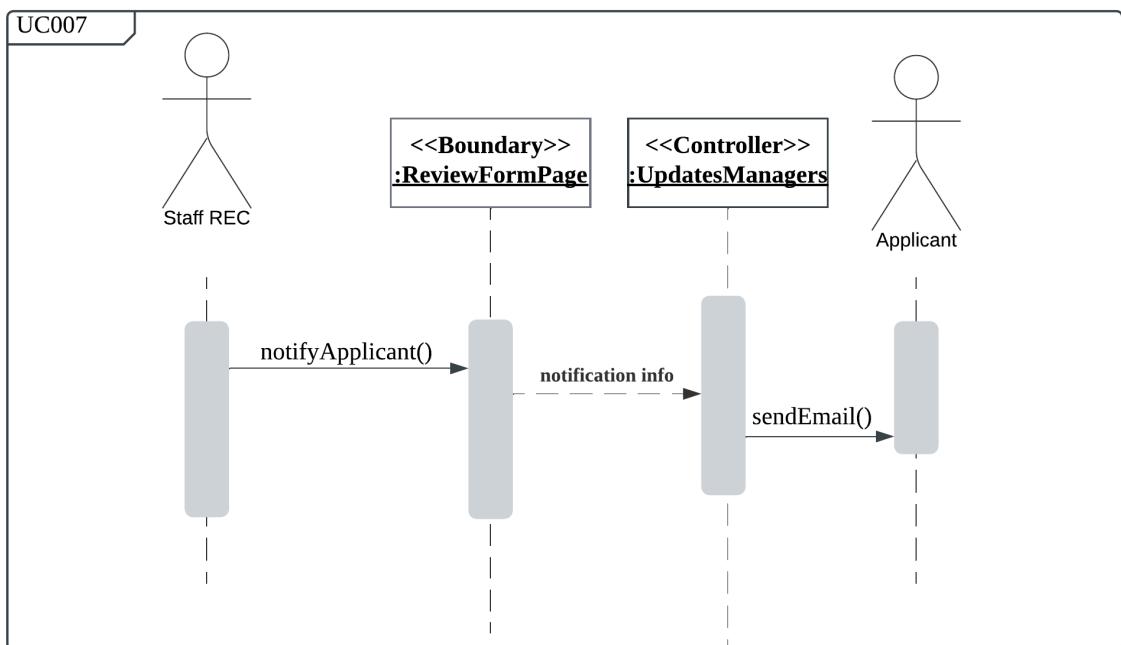


Figure 2.3.13: Sequence Diagram for Notify applicants

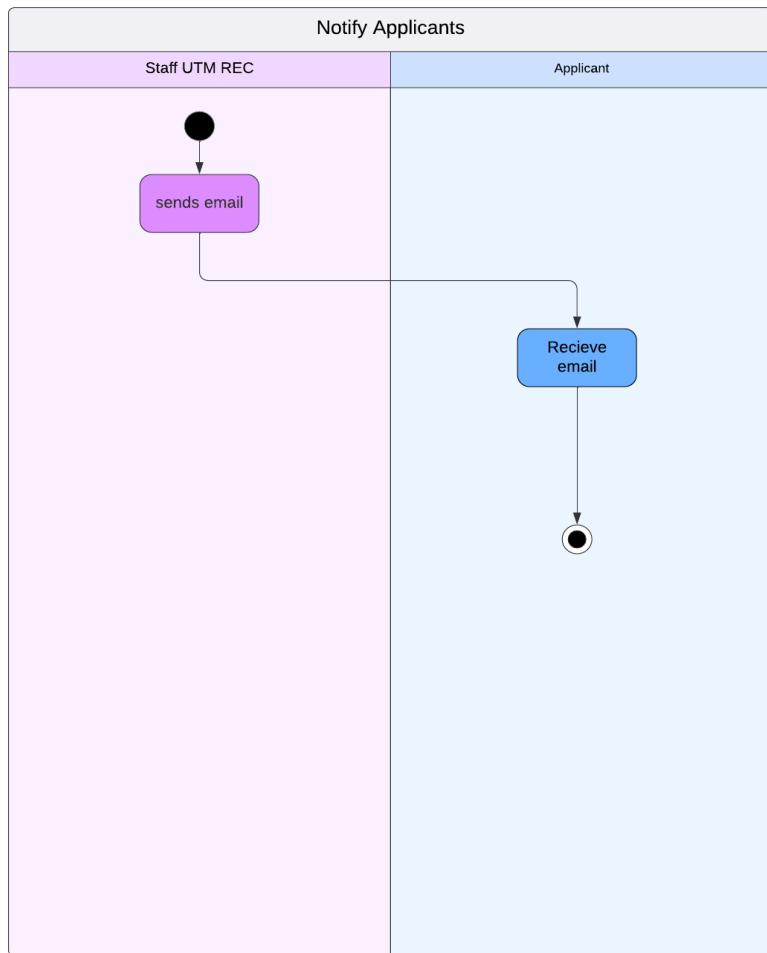


Figure 2.3.14: Activity Diagram for Notify applicants

2.3.8 UC008: Use Case Get help:

Table 2.8: Use Case Description for get help

Use case: Get help
ID: UC008
Actors: Staff REC Applicant
Preconditions:
Flow of events: 1. The applicant faced a problem and asked for help. 2. The system will try to address the applicant problem via chatbot. 3. If the chatbot could not solve the applicant problem, then 3.1 The system will forward the applicant problem to the appropriate staff REC 3.2 The staff REC will contact the applicant and solve the applicant's problem
Postconditions:
Alternative flow: 1. The staff REC could not help the applicant
Postconditions: 1. The staff REC arranges a meeting with the applicant.

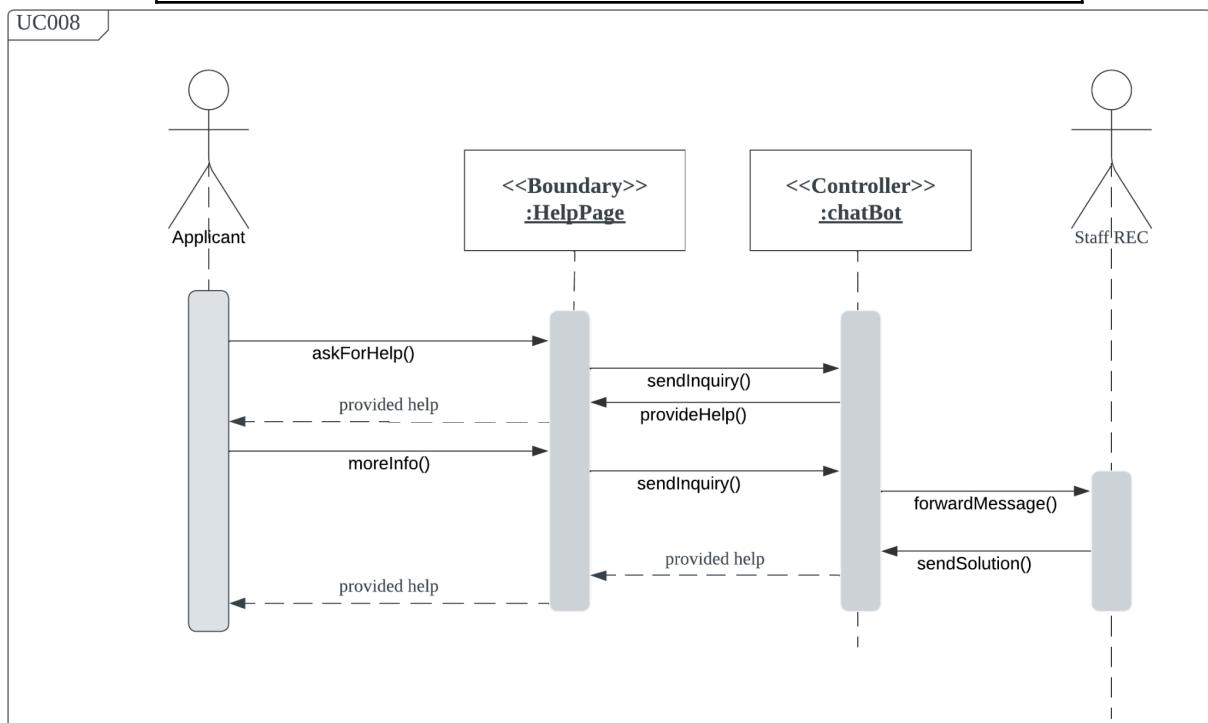


Figure 2.3.15: Sequence Diagram for Get help

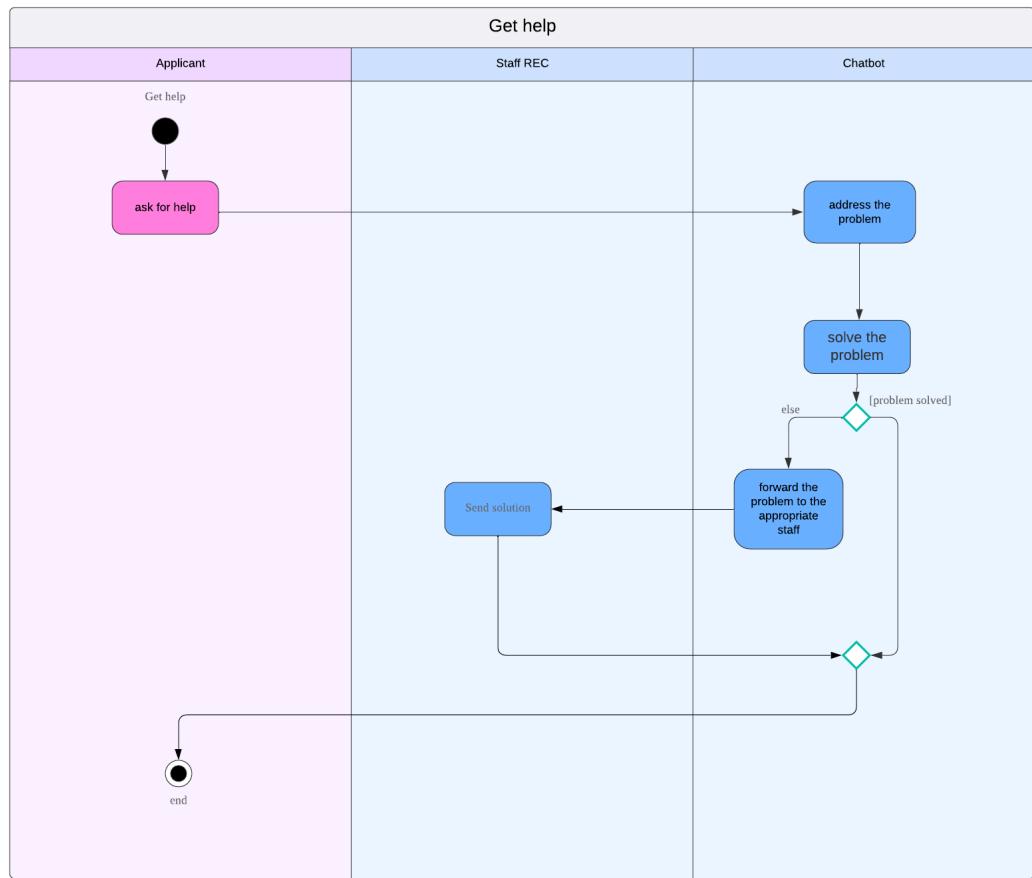


Figure 2.3.16: Activity Diagram for Get help

2.4 Performance and Other Requirements:

Software System Attributes:

- Usability: The system must be designed user-friendly to provide ease of use, featuring a clear and clean interface for all users.
- Reliability: The system must perform its functions accurately and consistently, minimizing errors or bugs.
- Maintainability: The system must be designed to be easily modified, repaired, and enhanced to accommodate future requirements or updates.
- Compatibility: The system must function seamlessly with Windows operating systems (Windows 10 and above) Android devices (version 10 and above) iOS devices (version 13 and above).

Performance:

- Response Time: The system should respond to user requests within a maximum of 5 seconds.
- Capacity: The system must support up to 500 concurrent user requests.

Other Requirements:

- Security: The system and its data must be protected from unauthorized access and malicious attacks to ensure confidentiality, integrity, and availability.
- Legal and Regulatory: The system must comply with all relevant laws, regulations, and standards applicable in Malaysia.

2.5 Design Constraints:

Below are some of the constraints that the UTM EAMS must adhere:

- **Technology stack constraints:** The system uses specific programming languages, frameworks and technologies such as Java, .NET, MySQL as mandated by the organization.
- **Deployment constraints:** The system must be deployable on UTM's existing infrastructure without requiring significant upgrades or additional servers.
- **Data retention constraints:** The system must store and retain user data for a predefined period (example: 8 years) before automatic deletion, as per institutional policies.
- **Localization constraints:** The system must support the Malay and English languages to cater to diverse users within Malaysia.
- **User environment constraints:** The system must operate efficiently in environments with limited internet bandwidth or intermittent connectivity.

3. Architectural Rationale

3.1 Architecture Style and Rationale

The chosen architectural pattern for our proposed system is the **Model-View-Controller (MVC)** architecture. This pattern was selected for several compelling reasons. Firstly, the system involves data that needs to be accessed through multiple views by different users. For instance, research information is accessible to applicants after submitting their research proposals. Applicants can make adjustments based on feedback from REC staff, who can also access the same research information to review and provide comments.

The three distinct layers in the MVC pattern significantly enhance the system's functionality by separating the presentation layer (view) from the data model (database). This separation enables the data model to evolve independently from the presentation, promoting flexibility and maintainability. Moreover, the MVC architecture supports the presentation of the same data in different formats, ensuring that changes made in one representation are seamlessly reflected across all views.

This approach ensures a modular, scalable, and efficient system capable of meeting diverse user needs while maintaining a high degree of adaptability.

4. Architectural Views

According to Madhavan Vivekanandan, software architecture views are structured representations of a software system's overall architecture, designed to be meaningful to various stakeholders. These views facilitate effective communication, enabling stakeholders to understand the architecture and confirm that the system will address their specific concerns.

The **4+1 Architecture View Model** is a widely adopted approach for documenting software architectures, offering a comprehensive perspective of the system through multiple viewpoints. This model consists of five key views: logical view, process view, development (implementation) view, physical (deployment) view and use cases (scenario) view. As shown in *Figure 4.1*, each view has specific usage and is represented by corresponding models or diagrams, as summarized in *Table 4.1*.

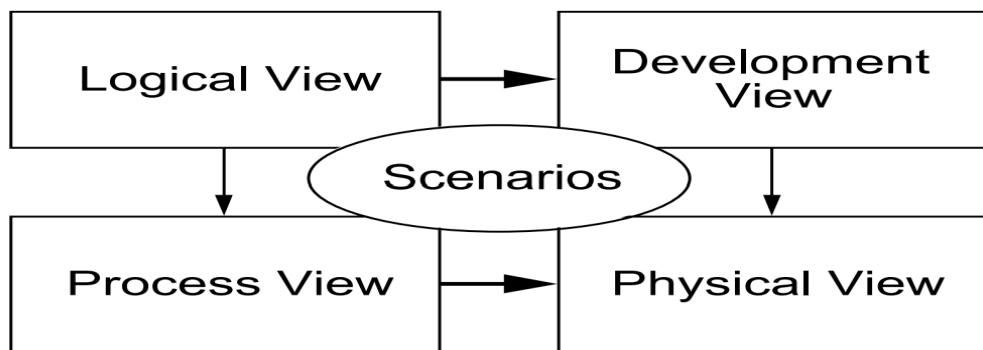


Figure 4.1 – Architectural Views

Table 4.1

Architectural Views	Description	Diagrams used
Use Case View	Highlights how the system meets stakeholder needs by presenting real-word use cases and scenarios.	Use Case Diagram
Logical View	Shows the key abstractions in the system as objects or object classes	Class Diagram
Process View	Illustrates the system's dynamic behavior, showing how the system is composed of interacting processes during runtime.	Sequence Diagram
Development (or Implementation) View	Shows how the software is decomposed for development	Component Diagram Package Diagram
Physical (or Deployment) View	Shows the system hardware and how software components are distributed across the processors in the system	Deployment Diagram

4.1 Use Case View

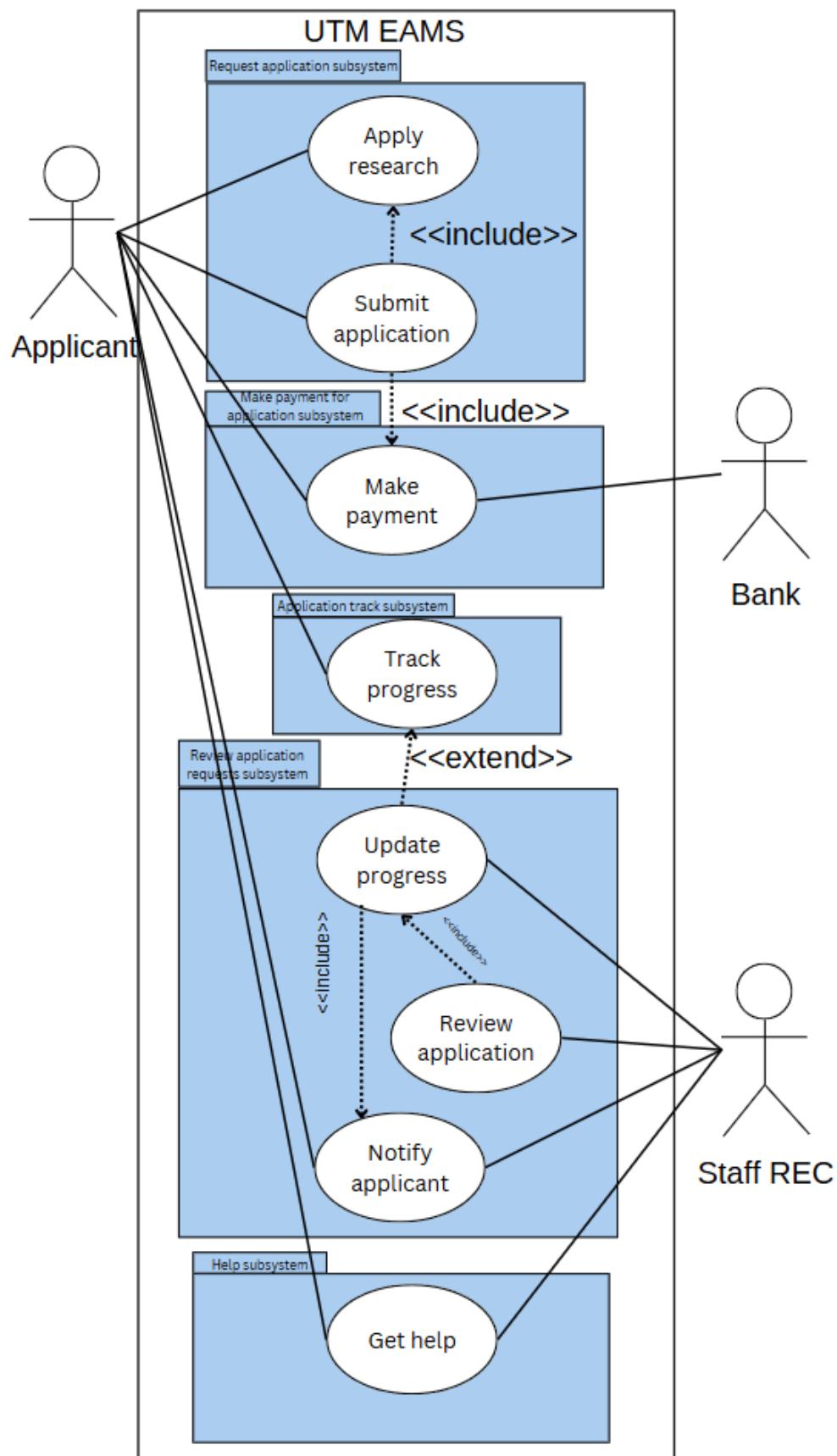


Figure 4.1.1: Use Case Diagram for <UTM EAMS>

4.2 Implementation View

This view shows the organization of the system's software modules in its development environment. It focuses on how the system is decomposed into components for implementation.

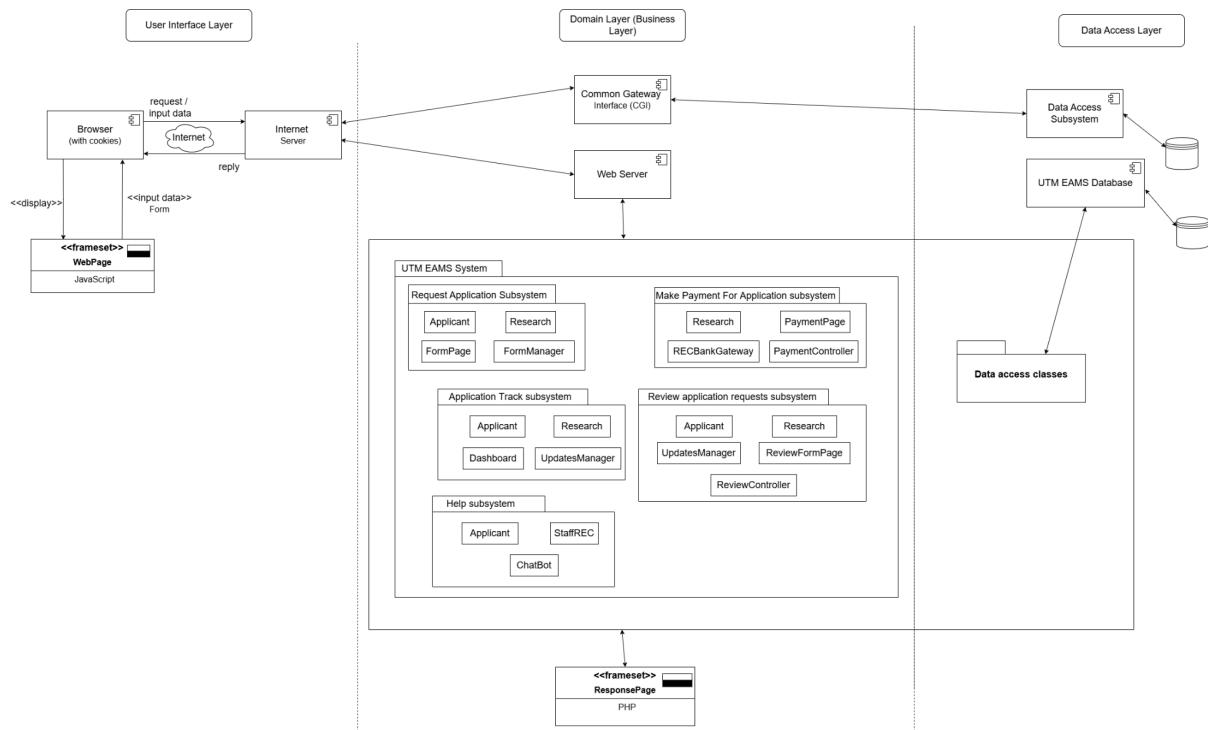


Figure 4.2.1: Component Diagram of <UTM EAMS>
https://drive.google.com/file/d/1zeSKS-RCSnXzU_OUApmx55V3m1WdqSZI/view?usp=sharing

The diagram illustrates the architecture of a high-level system for managing applications, payments and user interactions. The system is divided into four primary layers and a central data repository, each fulfilling distinct responsibilities and interconnected to achieve seamless functionality.

1. User Interface Layer:

- Components: FormPage, ResponsePage, Dashboard,
- Responsibilities: Collects user inputs and display outputs. It interacts with the users (applicants, staff) to gather requests, provide updates and facilitate payment-related actions.

2. Domain Layer (Business Layer):

- Components: FormManager, UpdatesManager, PaymentController, ReviewController.
- Responsibilities: Manages business logic and workflows. It processes applications, updates, reviews and payment transactions, connecting the user interface with the data layer.

3. Data Access Layer:

- Components: Data Access Classes, UTM EAMS Database
- Responsibilities: Manages interactions with the database. It stores and retrieves data related to applications, payments and updates.

4. Web Server:

- Components: Common Gateway Interface (CGI), Web Server.
- Responsibilities: Acts as mediator between the user interface and the domain layer by processing user requests and routing them to the appropriate components.

5. UTM EAMS Database:

- Responsibilities: Acts as the central repository for all system data, including applications, payments and updates.

Collaboration Between Subsystems:

- The User Interface Layer collects inputs and passes them to the Domain Layer, which processes the data using business logic.
- The Domain Layer interacts with the Data Access Layer to store and retrieve necessary data from the UTM EAMS Database.
- The Web Server coordinates the flow of information between the client-side (user) and the server-side (business logic and database).

Bidirectional arrows represent the flow of data and control signals between layers. This clear separation of responsibilities allows the system to process user requests, handle business logic and manage data efficiently, providing a scalable and secure platform for managing applications and payments.

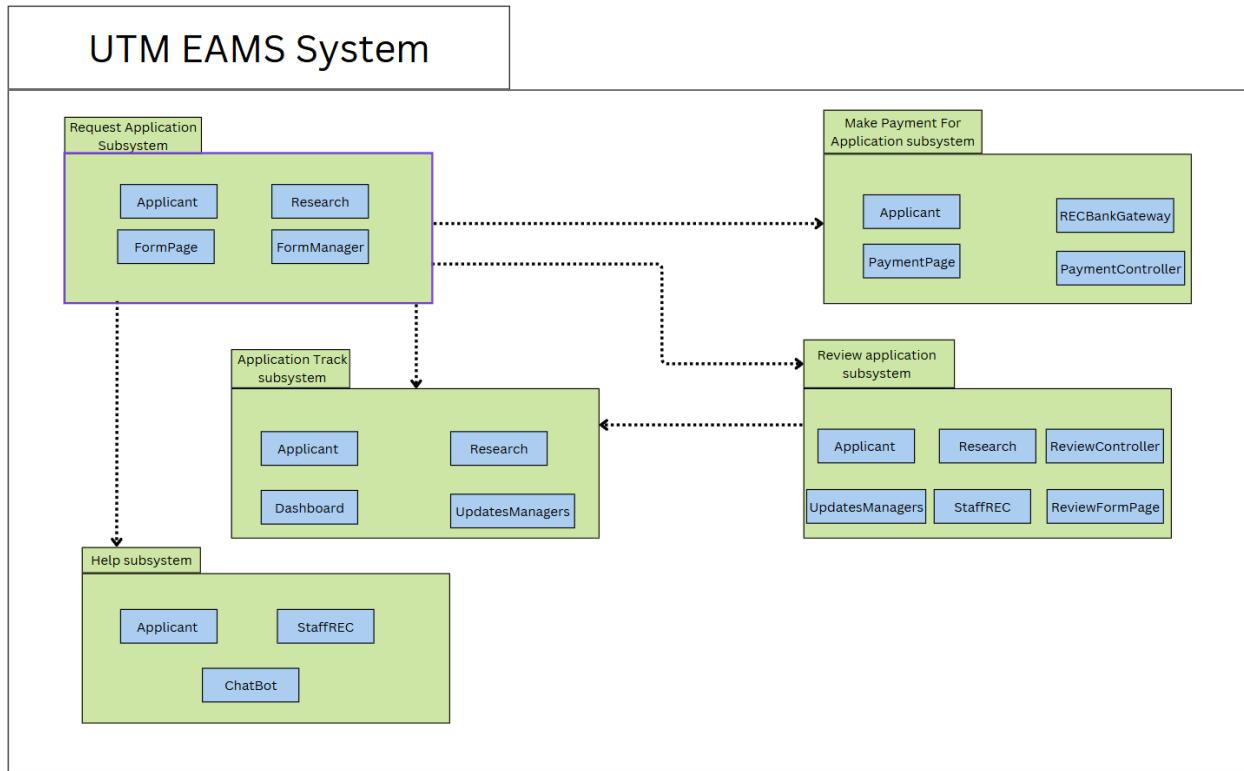


Figure 4.2.2: Package Diagram for <UTM EAMS>

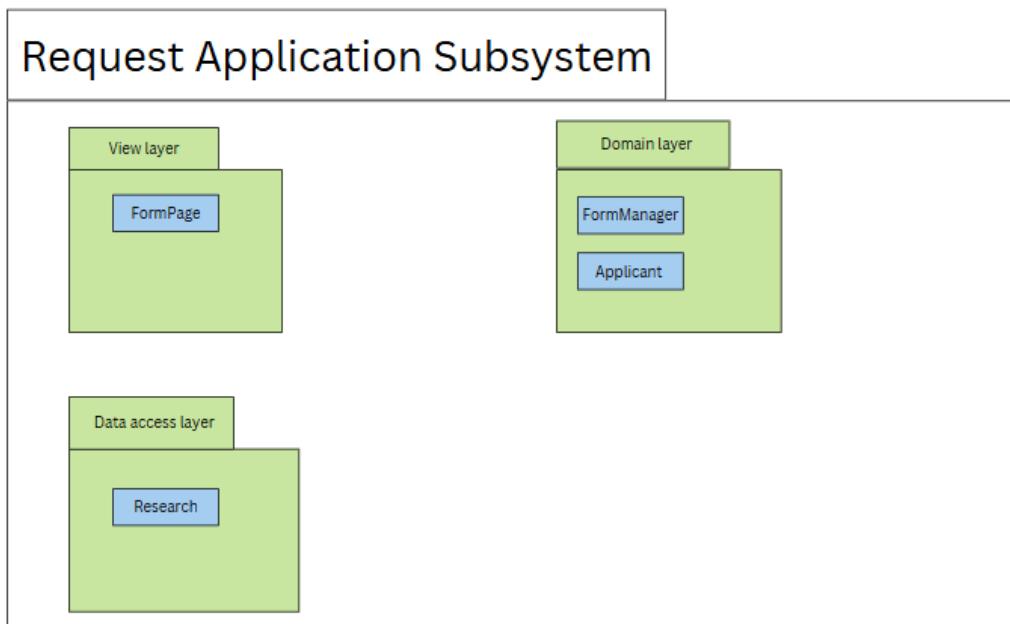


Figure 4.2.3: Package Diagram for <Request Application> Subsystem

Make Payment For Application subsystem

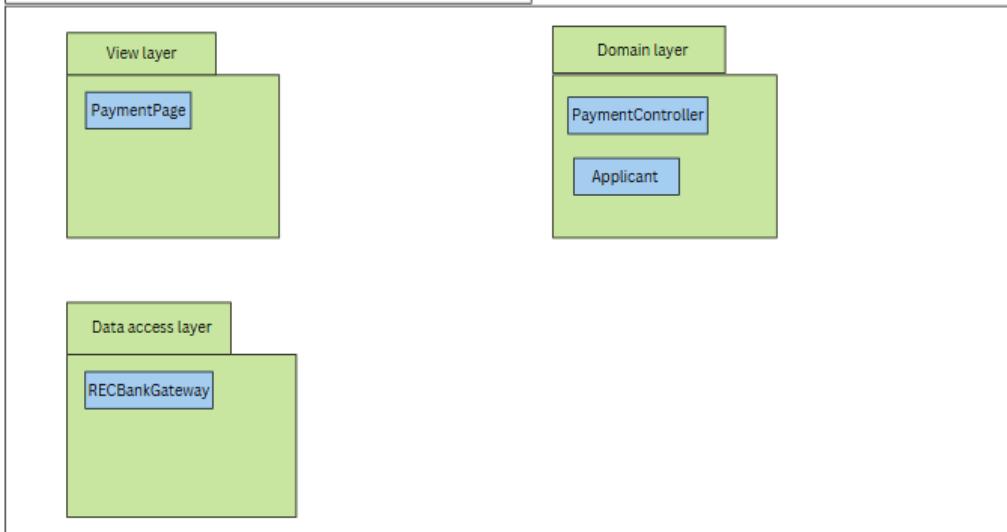


Figure 4.2.4: Package Diagram for <Make Payment for Application> Subsystem

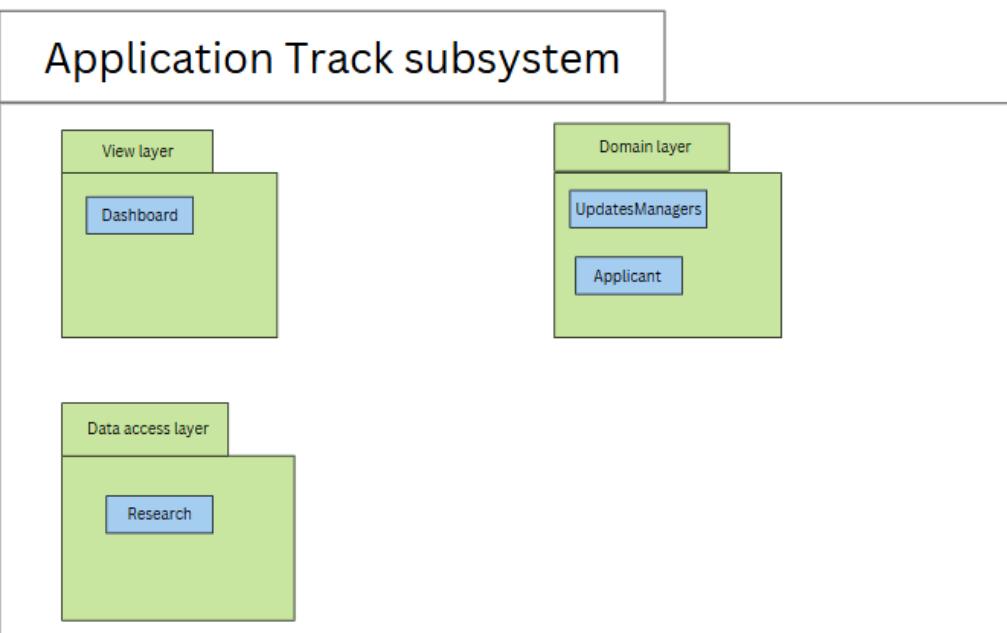


Figure 4.2.5: Package Diagram for <Application Track> Subsystem

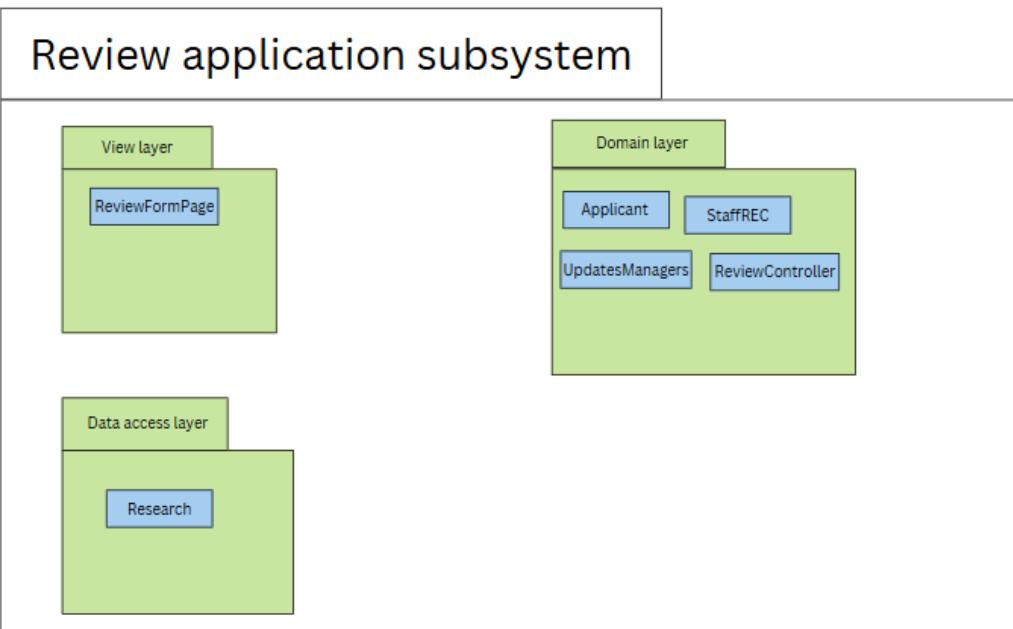


Figure 4.2.6: Package Diagram for <Review Application> Subsystem

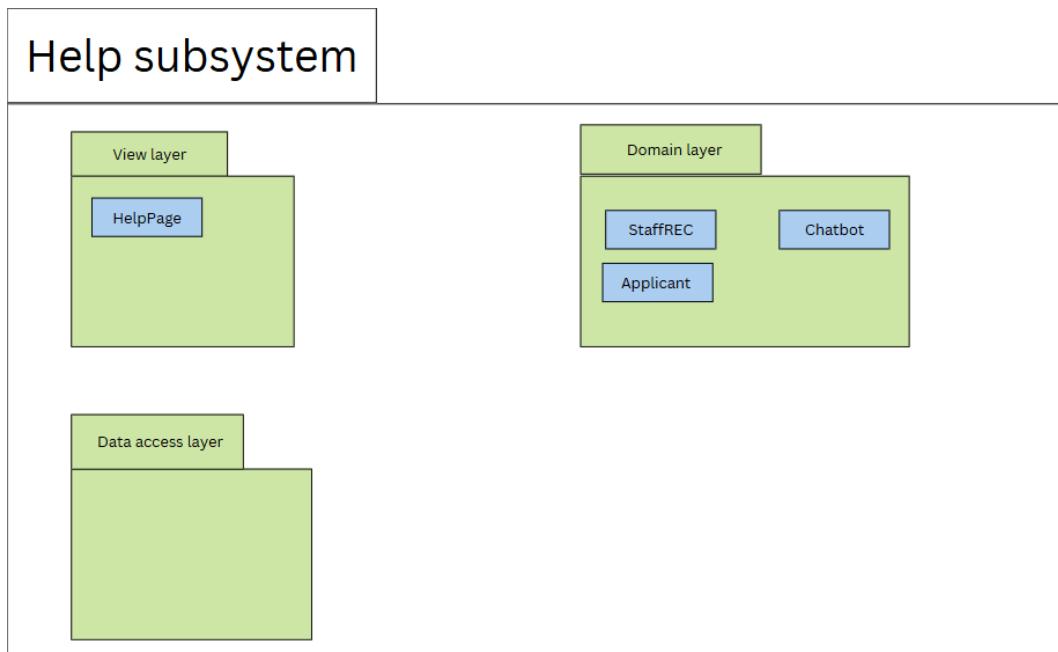


Figure 4.2.7: Package Diagram for <Help> Subsystem

4.3 Logical View

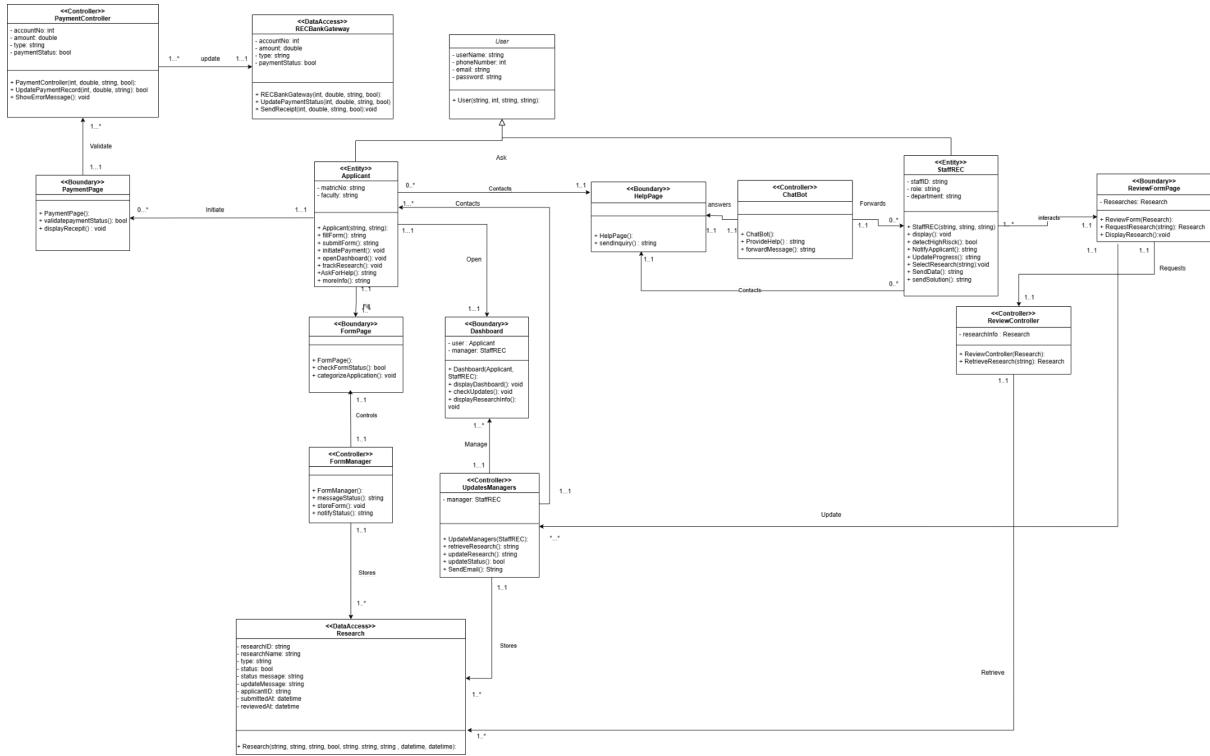


Figure 4.3.1: Class diagram for <UTM EAMS>
https://drive.google.com/file/d/1XBjGfDrWtb6WSM-y0OpAYBY6meJ_3Xl/view?usp=sharing

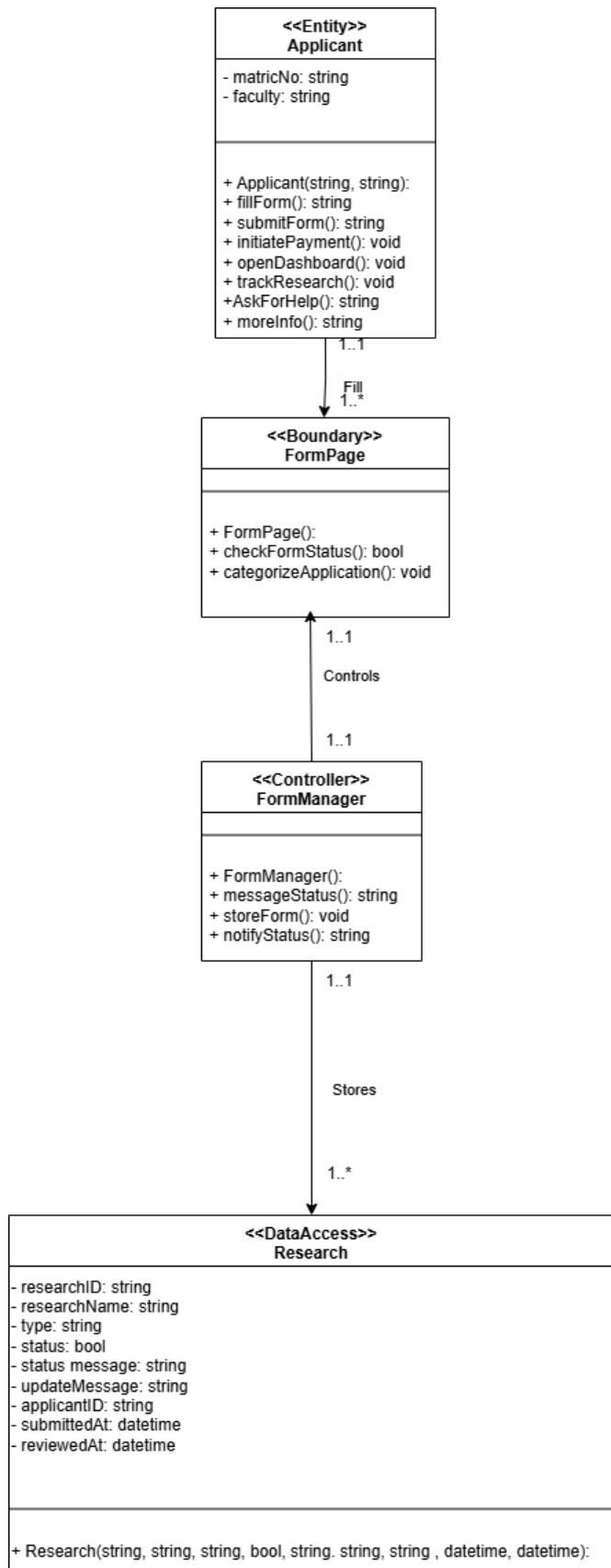


Figure 4.3.2: Class diagram for <Request Application> Subsystem

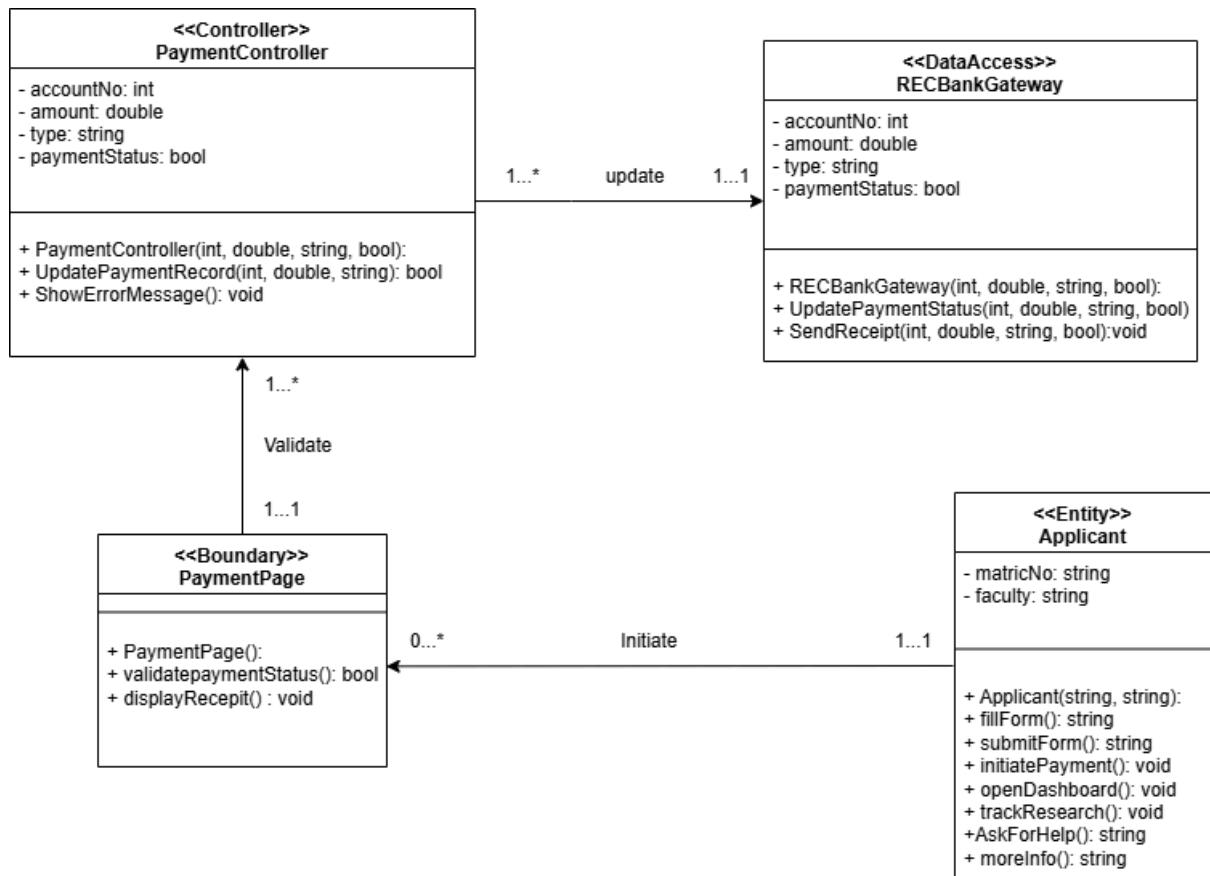


Figure 4.3.3: Class Diagram for <Make Payment for Application> Subsystem

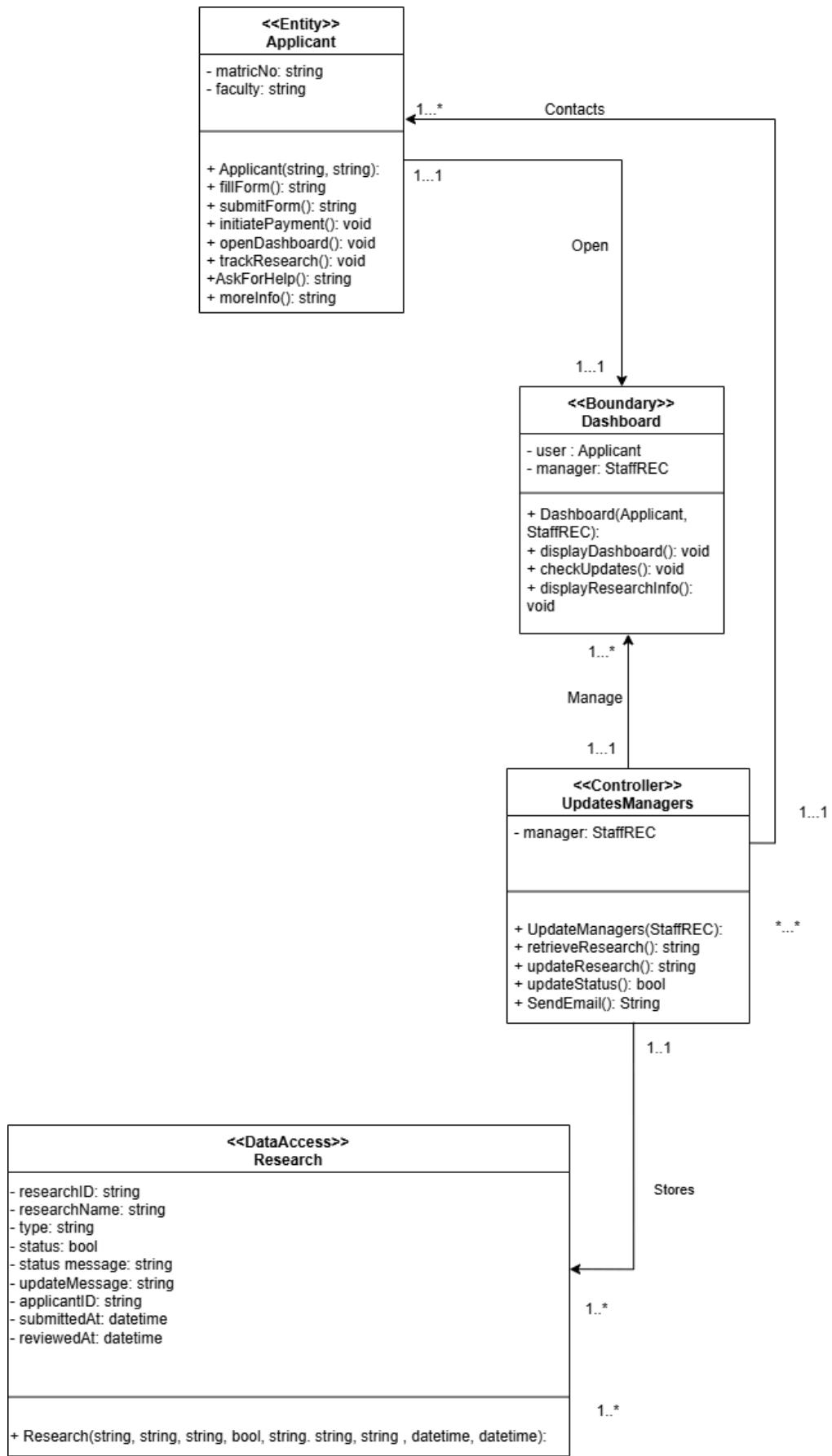


Figure 4.3.4: Class Diagram for <Application Track> Subsystem

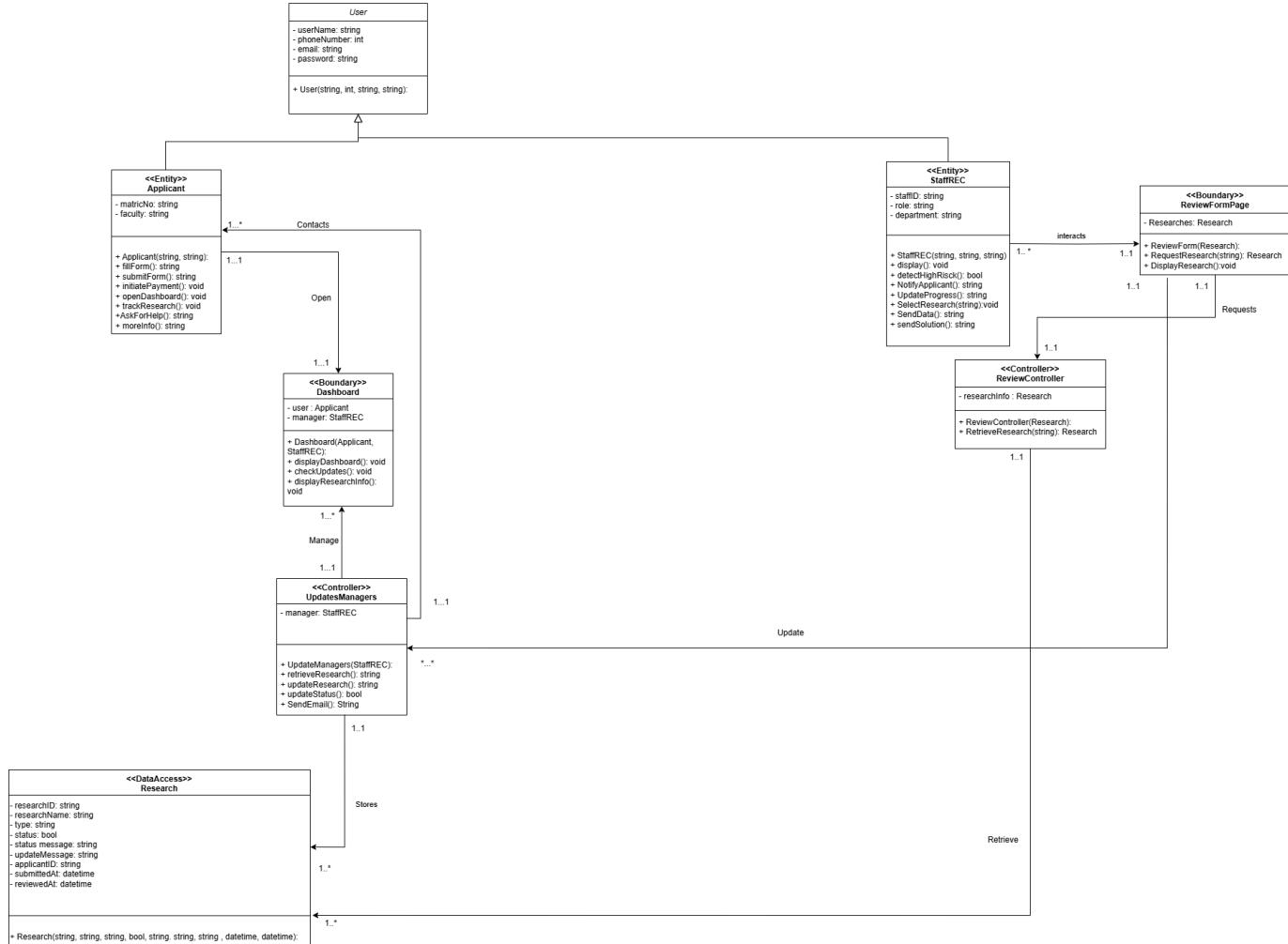


Figure 4.3.5: Class Diagram for <Review Application> Subsystem

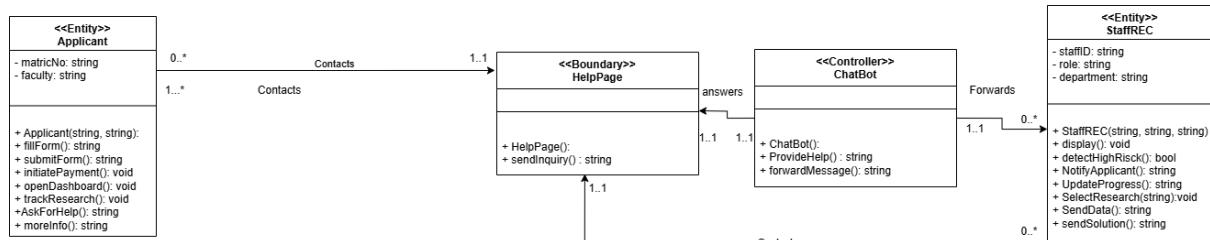


Figure 4.3.6: Class Diagram for <Help> Subsystem

4.4 Process View

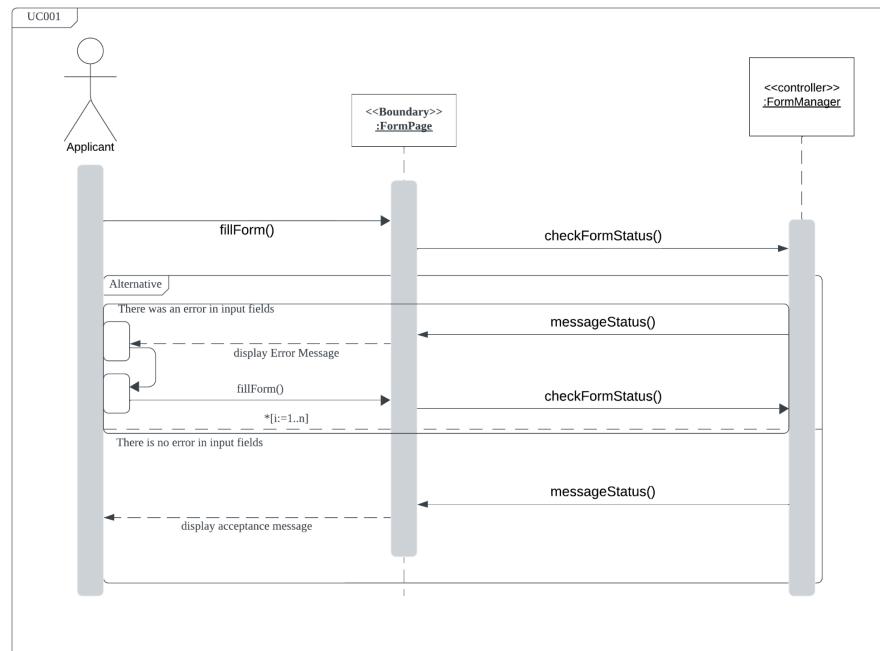


Figure 4.4.1: SD001 Sequence Diagram for <Apply Research Scenario>

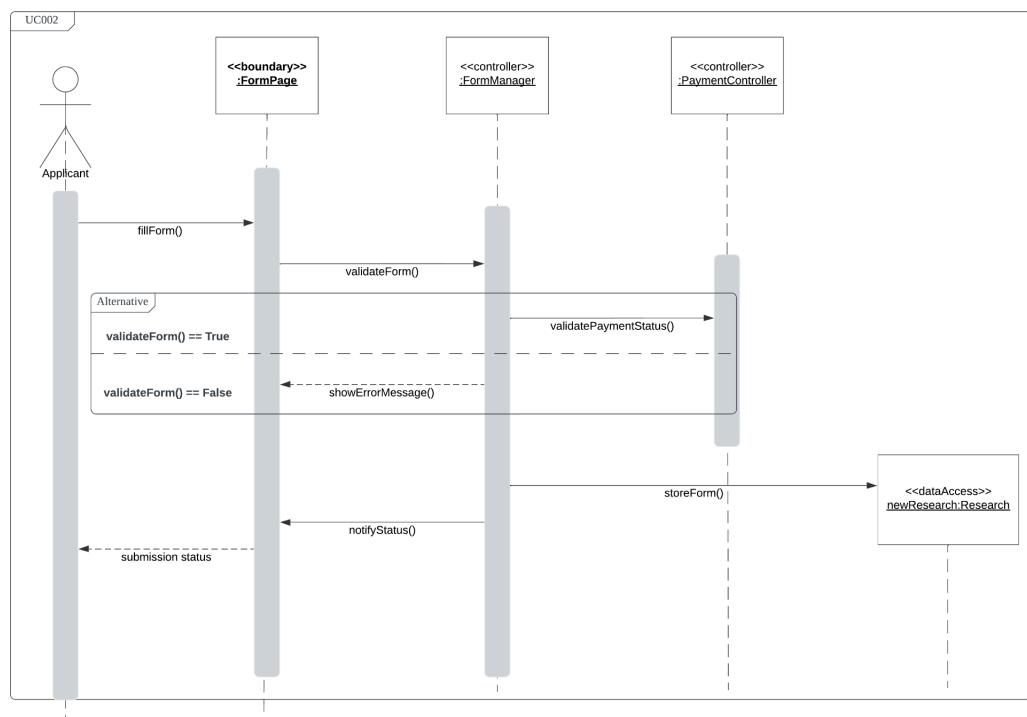


Figure 4.4.2: SD002 Sequence Diagram for <Submit Application scenario>

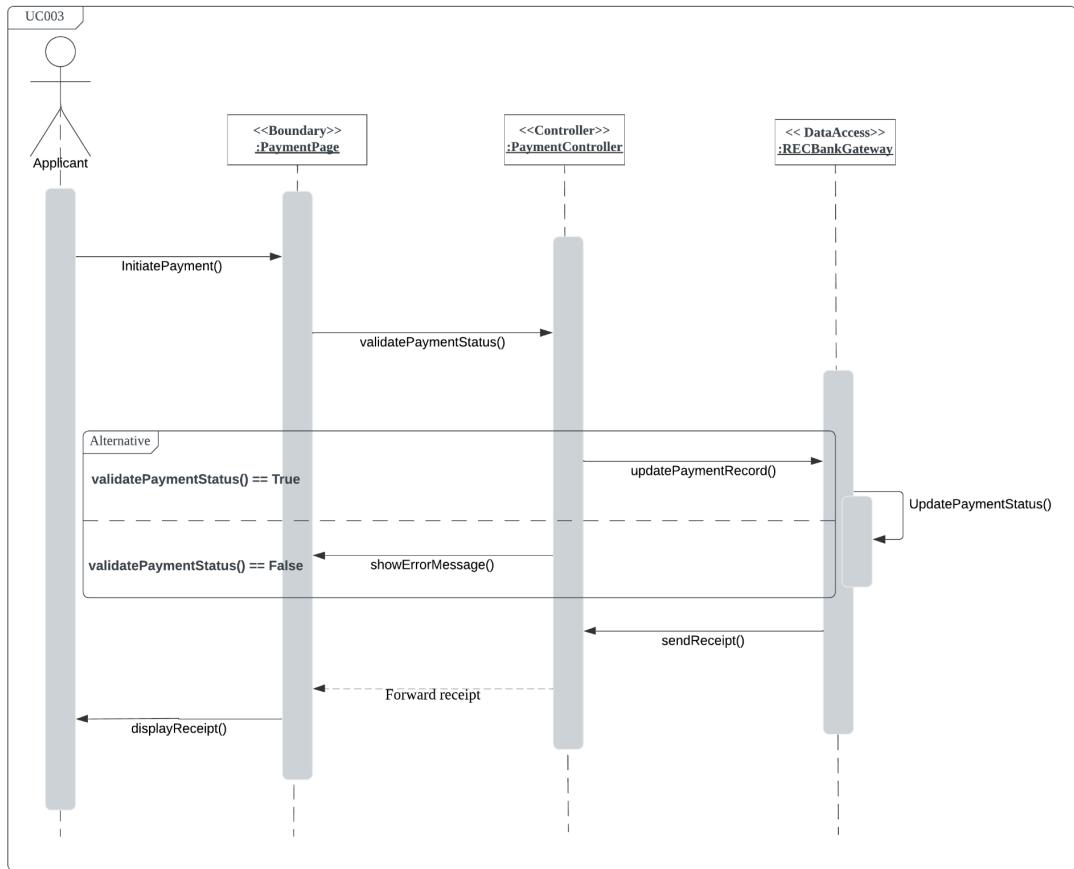


Figure 4.4.3: SD003 Sequence Diagram for <Make Payment for Application scenario>

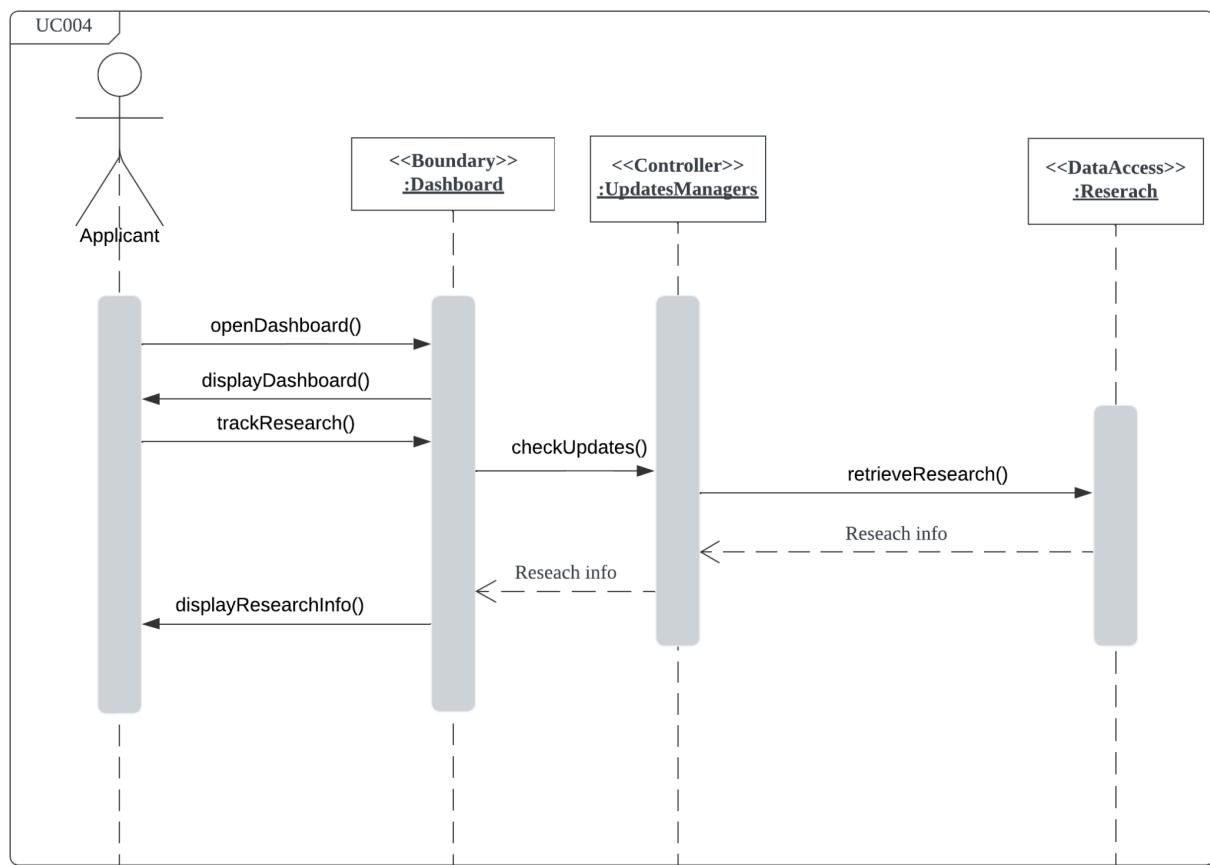


Figure 4.4.4: SD004 Sequence Diagram for <Track Progress scenario>

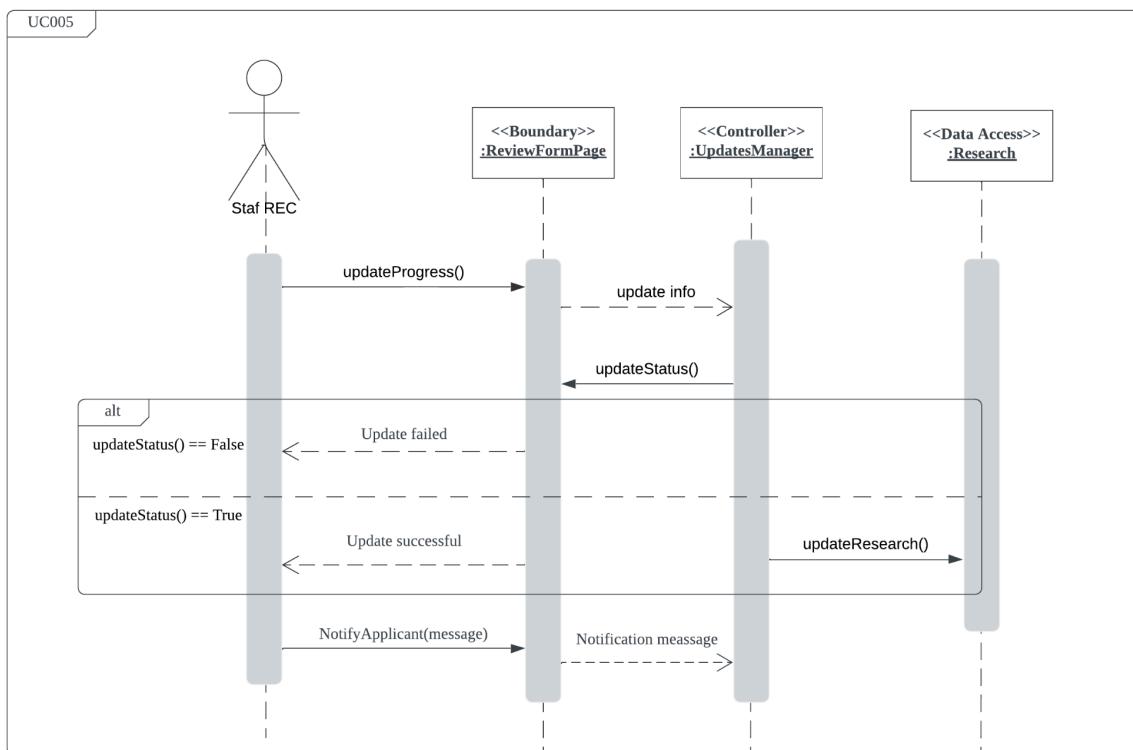


Figure 4.4.5: SD005 Sequence Diagram for <Update Progress scenario>

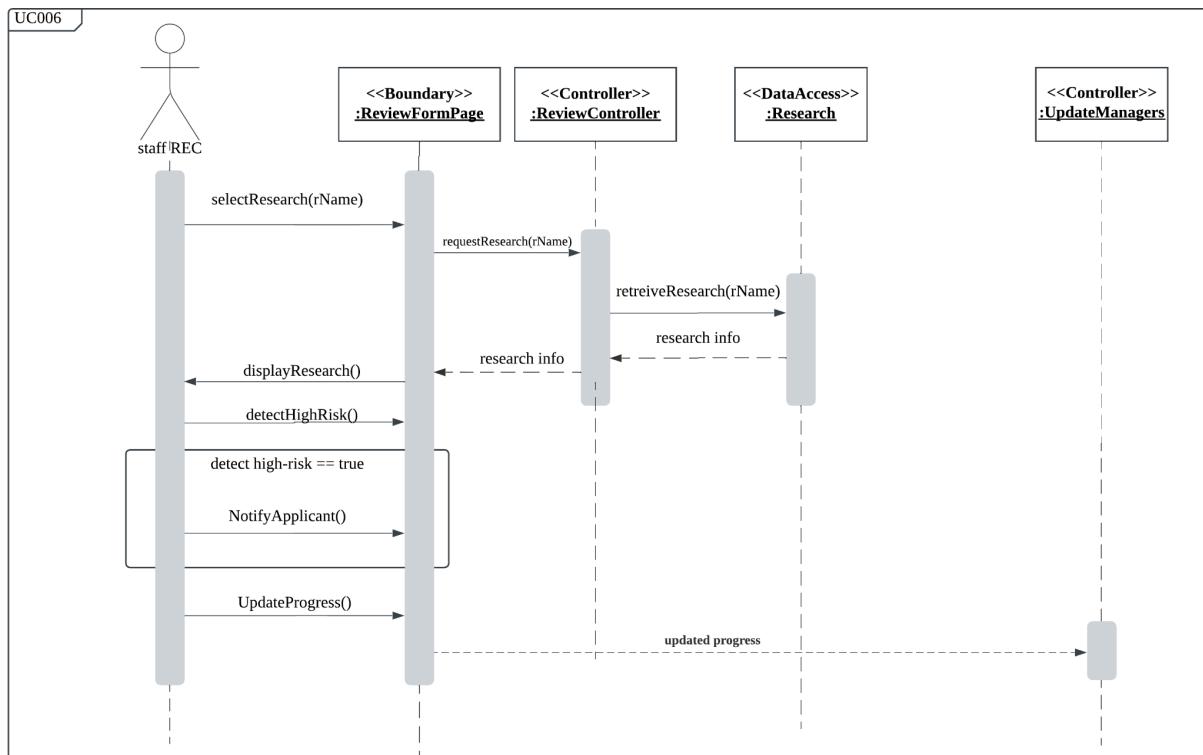


Figure 4.4.6: SD006 Sequence Diagram for <Review Application scenario>

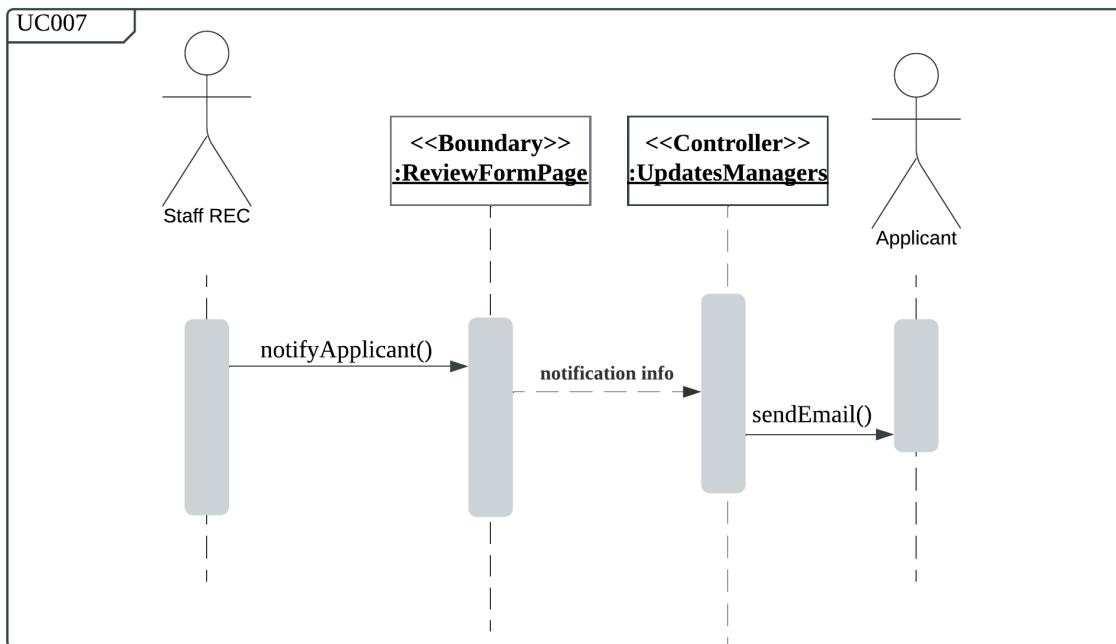


Figure 4.4.7: SD007 Sequence Diagram for <Notify Applicants scenario>

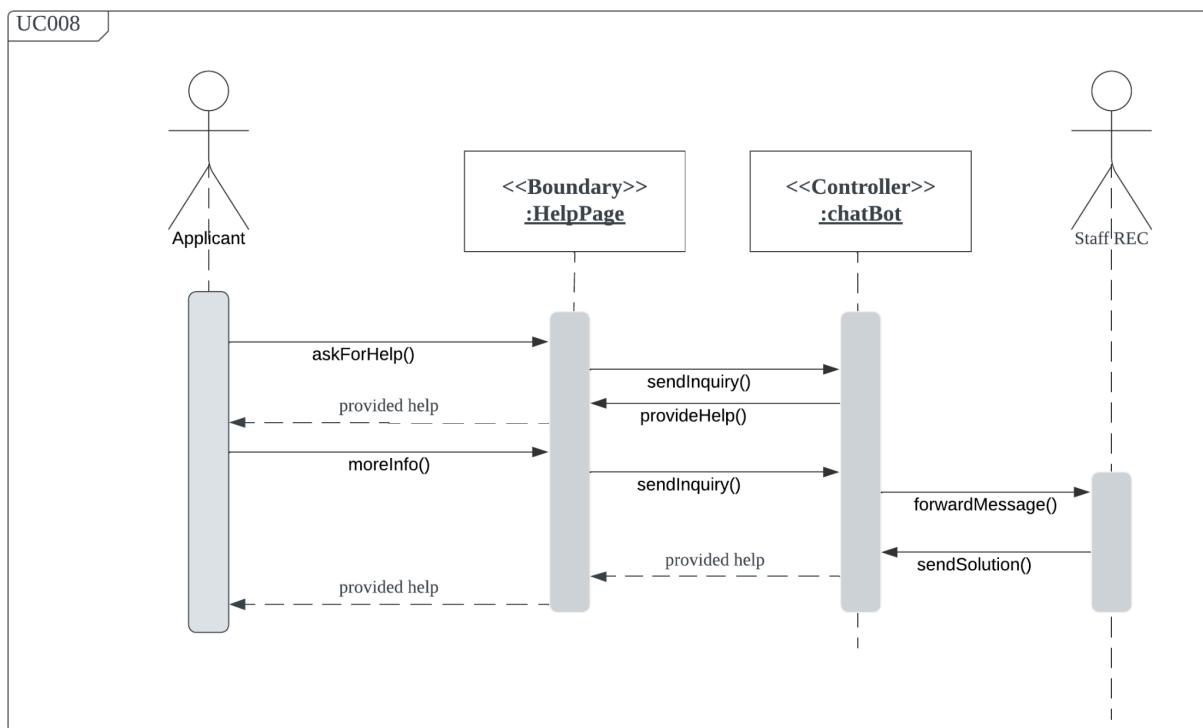


Figure 4.4.8: SD008 Sequence Diagram for <Get Help scenario>

4.5 Physical/ Deployment View

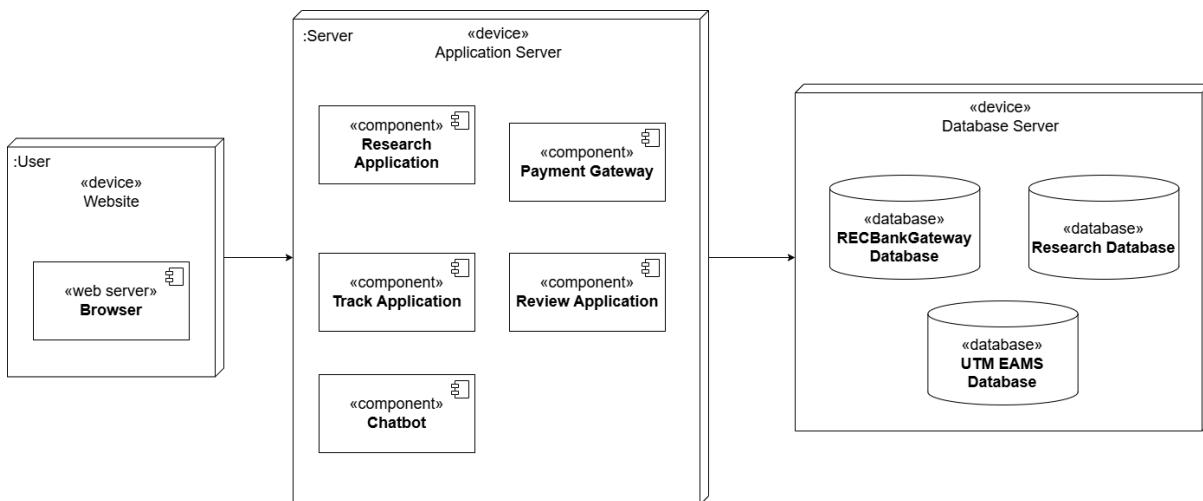


Figure 4.5: Deployment diagram for Internet-based System
https://drive.google.com/file/d/1eahgwYS7OH7CLY2fg9WtkGEAh_b5PGbG/view?usp=sharing

5. Data Design

The major data or systems entities are stored into a relational database named as UTM EAMS Database, processed and organized into n entities as listed in Table 5.1.

Table 5.1: Description of Entities in the Database

No.	Entity Name	Description
1.	Applicant	A derived entity from the general user entity that contains a user information, it fills the form page, initiates payments, opens the dashboard to check for progress, gets contacted by the UpdateManager, and uses the ChatBot for help or feedback.
2.	StaffREC	A derived entity from the general user entity that contains a StaffREC information, it interacts with the ReviewFormPage to retrieve researches, give Updates to the UpdateManager, and receives forwarded questions and feedback from ChatBot.
3.	ChatBot	A boundary that takes the Applicants' questions or feedback and forwards it to the StaffREC.
4.	Dashboard	A boundary, a web page that is opened(checked) by the applicants for updates, and managed(updated) by the Update manager.
5.	FormManager	A controller that controls(manages) the Dashboard and store research.
6.	FormPage	A boundary, a web page that is filled by applicants for submission and controlled(managed) by the FormManager.
7.	UpdatesManagers	A controller, it contains StaffREC information, it receives updates from the StaffREC, Contacts the applicants for the updates, manages the dashboard and stores research.
8.	Research	Data Access that contains all information about a research, the research information is stored by the FormManager and UpdatesManagers, the research information is retrieved by the ReviewController.
9.	PaymentPage	A boundary, A web payment page that is initiated by the applicant and validated by the payment controller.
10.	PaymentController	A controller contains payment information of an applicant, it validates the Payment of the payment page, and updates the information to the RECBankGateway.
11.	RECBankGateway	a Data access that contains applicant payment's information, and the payment information is updated by the payment controller.
12.	ReviewFormPage	a boundary that holds a research information, it gets interacted(triggered) by the StaffREC to request a research information from the ReviewController.

13.	ReviewController	a controller that holds a research information, it receives a request from the ReviewFormPage to Retrieve a research information from the data access Research.
-----	------------------	---

5.1 Data Dictionary

5.1.1 Entity: <Applicant>

Attribute Name	Type	Description
matricNo	VARCHAR(10)	matric number of an applicant.
Faculty	VARCHAR(20)	faculty of an applicant.

5.1.2 Entity: <StaffREC>

Attribute Name	Type	Description
staffID	VARCHAR(10)	the identification number of a StaffREC.
role	VARCHAR(10)	The position of a StaffREC.
department	VARCHAR(20)	the department of a StaffREC.

5.1.3 Entity: <ChatBot>

No attributes

5.1.4 Entity: <Dashboard>

Attribute Name	Type	Description
user	Applicant	store the user information.
manager	StaffREC	store StaffREC staff information.

5.1.5 Entity: <FormManager>

No attributes

5.1.6 Entity: <FormPage>

No attributes

5.1.7 Entity: <UpdatesManagers>

Attribute Name	Type	Description
Manager	StaffREC	Store the information of a StaffREC.

5.1.8 Entity: <Research>

Attribute Name	Type	Description
researchID	VARCHAR(10)	The identification number of a Research.
researchName	VARCHAR(10)	The name of a Research.
type	VARCHAR(10)	The type of a Research.
status	bool	The state of the research
statusMessage	VARCHAR(10)	The status statement that a research display.
updateMessage	VARCHAR(50)	The update message a research display.
applicantID	VARCHAR(10)	The identification of the applicant who submitted the research.
submittedAt	datetime	The date and time of submission.
reviewedAt	datetime	The date and time of the revision.

5.1.9 Entity: <PaymentPage>

No attributes

5.1.10 Entity: <PaymentController>

Attribute Name	Type	Description
accountNo	int	The account number of the applicant who performed the payment.
amount	double	The amount the applicant paid.
type	VARCHAR(15)	The type of payment by the applicant.
paymentStatus	bool	The status of the payment.

5.1.11 Entity: <RECBankGateaway>

Attribute Name	Type	Description
accountNo	int	The account number of the applicant who performed the payment.
amount	double	The amount the applicant paid.
type	VARCHAR(15)	The type of payment by the applicant.
paymentStatus	bool	The status of the payment.

5.1.12 Entity: <ReviewFormPage>

Attribute Name	Type	Description
Researches	Research	the information of the research requested from the ReviewController by the StaffREC

5.1.13 Entity: <ReviewController>

Attribute Name	Type	Description
researchInfo	Research	the information of the research retrieved from the Data Access Research and requested ReviewController.

6. User Interface Design

The interface includes the different functionalities that the user needs in order to apply successfully to his ethics Application. The new design covers all of the old google form application including the upload of different requirements asked from the user.

6.1 Screen Images



Copyright ©2024 Universiti Teknologi Malaysia

Figure 6.1 For <Apply Research>



Figure 6.2 For Login

Figure 6.3 For <Track Progress>


UTM
 UNIVERSITI TEKNOLOGI MALAYSIA

Department of
 Deputy Vice-Chancellor
 (Research & Innovation)



Copyright ©2024 Universiti Teknologi Malaysia

Payment method

Card information

0000 0000 0000 0000	MM/YY	CVC
Name		

Next →

Figure 6.4 For <Make Payment>


UTM
 UNIVERSITI TEKNOLOGI MALAYSIA

Department of
 Deputy Vice-Chancellor
 (Research & Innovation)



Has data collection for your research begun?

Yes

No

Type of Research Ethics Application:

Clinical

Non-Clinical

Animal

Next →

Copyright ©2024 Universiti Teknologi Malaysia

Figure 6.5 For <Track Progress>

+6 07-553 0357
dvcri@utm.my
Contact 



Department of
Deputy Vice-Chancellor
(Research & Innovation)

Upload your application form



select your file or drag and drop

[browse](#)

Upload your C.V



select your file or drag and drop

[browse](#)

Upload your Research Gantt Chart



select your file or drag and drop

[browse](#)

[Next →](#)

Copyright ©2024 Universiti Teknologi Malaysia

Figure 6.6 For <Update Progress>

+6 07-553 0357
dvcri@utm.my
Contact 



Department of
Deputy Vice-Chancellor
(Research & Innovation)

Upload your Research Protocol



select your file or drag and drop

[browse](#)

Upload Proof of Protocol Validation



select your file or drag and drop

[browse](#)

Upload your Consent Letter



select your file or drag and drop

[browse](#)

[Next →](#)

Copyright ©2024 Universiti Teknologi Malaysia

Figure 6.7 For <Update Progress>

 +6 07-553 0357  dveri@utm.my 


UTM
 UNIVERSITI TEKNOLOGI MALAYSIA

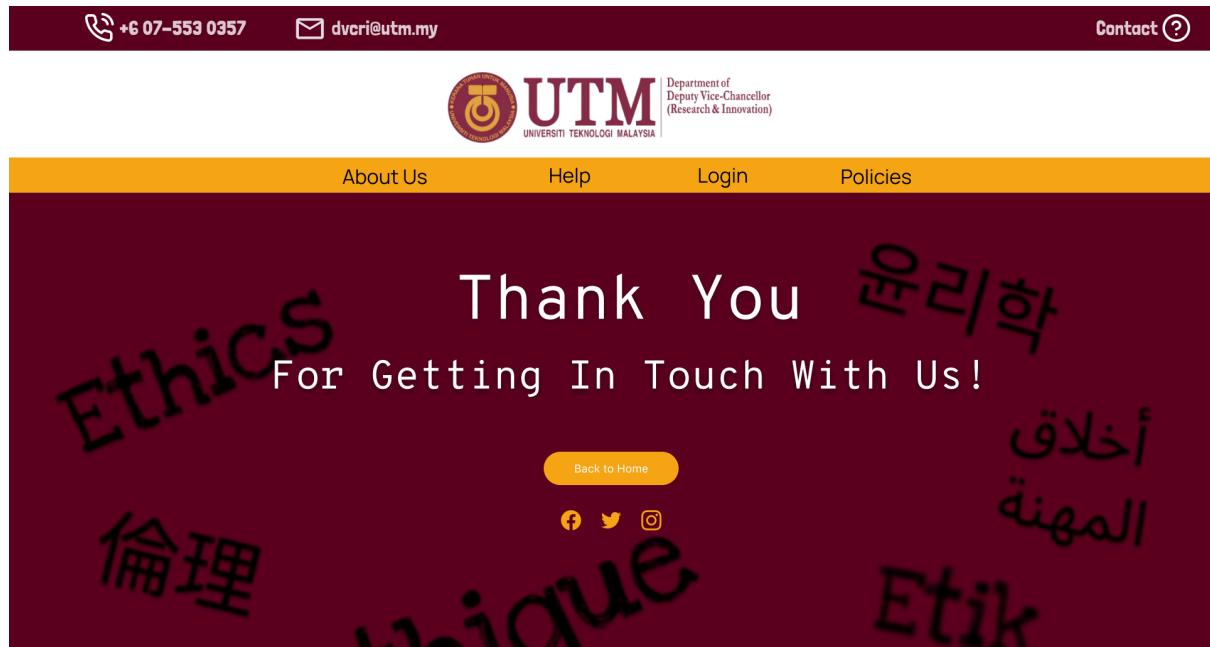
Department of
 Deputy Vice-Chancellor
 (Research & Innovation)

Upload Proof of Payment <div style="border: 1px solid #ccc; width: 100%; height: 30px; margin-top: 10px;"></div> <div style="border: 1px solid #ccc; width: 100%; height: 30px; margin-top: 10px;"></div> <div style="text-align: center; margin-top: 5px;">  select your file or drag and drop </div> <div style="text-align: center; margin-top: 10px;"> browse </div>	Upload Checklist Form <div style="border: 1px solid #ccc; width: 100%; height: 30px; margin-top: 10px;"></div> <div style="border: 1px solid #ccc; width: 100%; height: 30px; margin-top: 10px;"></div> <div style="text-align: center; margin-top: 5px;">  select your file or drag and drop </div> <div style="text-align: center; margin-top: 10px;"> browse </div>	Other Related Documents <div style="border: 1px solid #ccc; width: 100%; height: 30px; margin-top: 10px;"></div> <div style="border: 1px solid #ccc; width: 100%; height: 30px; margin-top: 10px;"></div> <div style="text-align: center; margin-top: 5px;">  select your file or drag and drop </div> <div style="text-align: center; margin-top: 10px;"> browse </div>
--	--	--

[Finish Attempt →](#)

Copyright ©2024 Universiti Teknologi Malaysia

Figure 6.8 For <Submit Application>



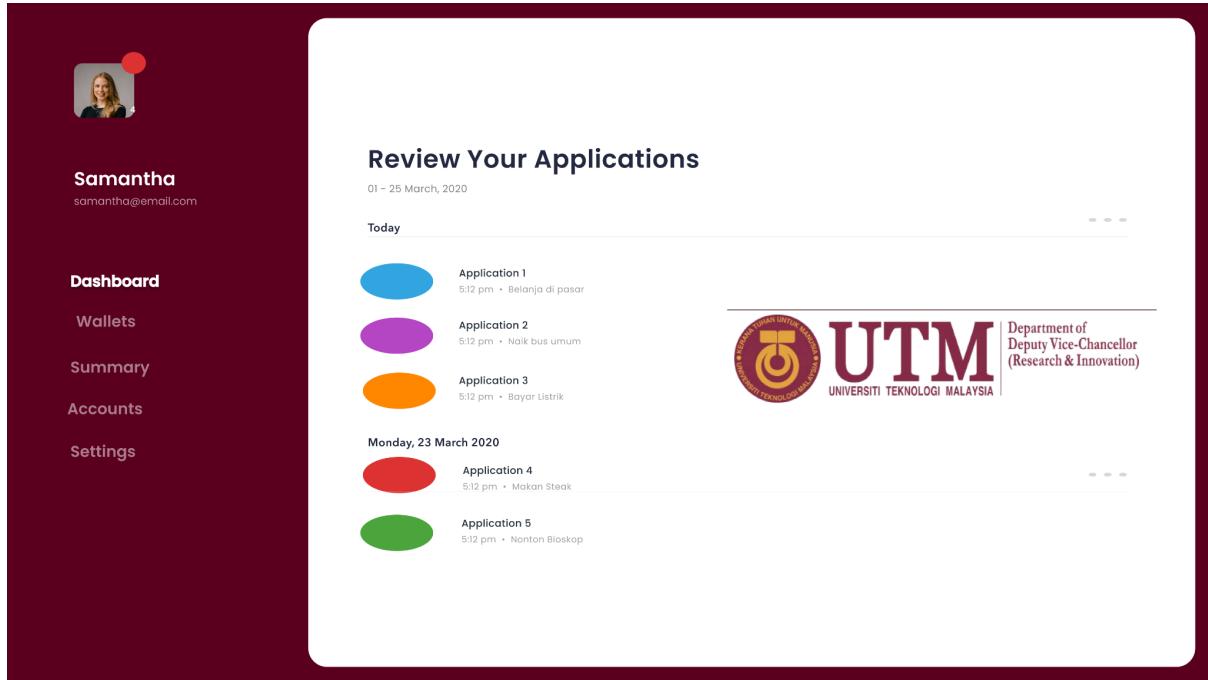


Figure 6.10 For <Review Application>

The screenshot shows a help center page with a yellow header containing 'Home', 'Dashboard', 'Resources', 'Pricing', 'Log in', and 'Sign up'. The main title is 'Ask us anything' with a 'FAQs' link above it. A search bar is present. The UTM logo is on the right.

Three frequently asked questions are listed:

- How do I change my account email?**
You can log in to your account and change it from your Profile > Edit Profile. Then go to the general tab to change your email.
- What should I do if my payment fails?**
If your payment fails, you can use the COD payment option, if available on that order. If your payment is debited from your account after a payment failure, it will be credited back within 7-10 days.
- What is your cancellation policy?**
You can now cancel an application while it is in processing status. Any amount paid will be credited into the same payment mode using which the payment was made.

A 'Still have questions?' button with a 'Get in touch' link is at the bottom.

Copyright © 2024 Universiti Teknologi Malaysia is at the bottom.

Figure 6.11 For <Get Help>

7. Traceability

Traceability between sequence diagrams for each use case vs. corresponding class diagrams (entities) are listed as in *Table 7.1*. The table establishes a clear and consistent connection between system behaviors (dynamic interactions) and components (static structures), facilitating efficient tracing of related behaviors and attributes. It improves maintainability, makes dependency visualization and analyzing easier, and makes debugging and future scaling of the system simpler.

Table 7.1: Requirement Matrix

Package, UC, Sequence Diagram / Entity Class	Applicant	StaffREC	Research	Dashboard	ChatBot	RECBankGateway
P001, UC001, SD001	X					
P001, UC002, SD002	X		X			
P002, UC003, SD003	X					X
P003, UC004, SD004	X		X	X		
P003, UC005, SD005		X	X			
PC004, UC006, SD006		X	X	X		
PC004, UC007, SD007	X	X	X	X		
PC005, UC008, SD008	X	X			X	