**Jordan University of Science and Technology**
**College of Computer Sciences & Information Technology**

# JUST Exam Reservation System

*A project submitted*
*in partial fulfillment of the requirements for the degree of*
*Bachelor's in software*
*engineering*

**By**

Sulaf Rafat Zayed Saeed Abu-Osbeh

Omar Khaled Mohammad Al-Dwairy

**Supervised by**

Mahmoud Mohammad Khair Hammad

**Committee Member Names**

Zakarea Mohammad Hussain Al-Shara

Hasan Al-Bzoor

**January 6th**
**2025**

# UNDERTAKING

This is to declare that the project entitled “JUST Exam Reservation System” is an original work done by undersigned, in partial fulfillment of the requirements for the degree “bachelor’s in software engineering” at the Software Engineering Department, College of Computer and Information Technology, Jordan University of Science and Technology.

All the analysis, design and system development have been accomplished by the undersigned. Moreover, this project has not been submitted to any other college or university.

Student 1: Sulaf Rafat Zayed Saeed Abu-Osbeh signature: Sulaf

Student 2: Omar Khaled Mohammad Al-Dwairy signature: Omar

# ABSTRACT

## Descriptive

This project template aimed to collect and specify the requirements for JUST's exam reservation system. The team was assembled to address the challenges of the current system, which makes the process of exam reservation turbulent, somewhat challenging, and time-consuming. Driven by the desire to create a more stable, user-friendly, and efficient system, we decided to collect the requirements of the existing system and incorporate additional functionalities to better align it with the evolving needs of the university and its students.

**Motivation:** To develop a stable, modern system that complies with current IT standards in the market while effectively fulfilling its purpose.

**Scope:** The scope of usage, as well as requirements determination and collection, extends across the departments of JUST's E-Learning Center and the Center of Computer and Information Systems.

**Methods:** We paid close attention to the interface and database of the current system to understand its operation and the fluidity of the existing exam reservation process. After examining the system, we gathered input and identified desired requirements from current system users and stakeholders.

The process of examining the system firsthand, communicating with users, identifying issues and weaknesses, and pinpointing potential improvements enabled us to design diagrams for the new system based on the existing one. Furthermore, it helped us determine the tools required to bring the improved system to fruition.

# ACKNOWLEDGMENT

We first thank God for His might and care upon us, as He is the One who takes care of us and lights our way: (يَرْفَعِ اللَّهُ الَّذِينَ آمَنُوا مِنْكُمْ وَالَّذِينَ أُوتُوا الْعِلْمَ دَرَجَاتٍ وَاللَّهُ بِمَا تَعْمَلُونَ خَبِيرٌ ﴿ [المجادلة: 11]

We then extend our deepest gratitude and appreciation to our families, who supported us and stood by us throughout this journey. To our mothers and fathers, especially, who carried us through the ups and downs and challenges of life—may God protect them, grant them happiness, and bless them with peace, gratitude, and good health.

We also hold a special place in our hearts for Professor Mahmoud Hammad who guided us during this project.

Finally, we express our gratitude to our professors at the Software Engineering Department at JUST, who have provided us with both knowledge and life lessons over the past four years.

# CONTENT

# LIST OF TABLES

# LIST OF FIGURES

# LIST OF ACRONYMS AND ABBREVIATIONS

| | |
|---|---|
| API | Application Programming Interface |
| REST | Representative State Transfer WS |
| RPC | Remote Procedure Call |

# CHAPTER 1: INTRODUCTION

## 1.1 Overview

This system is designed to facilitate the registration and management of all semester-related exams, including but not limited to First Exams, Second Exams, Mid Exams, Final Exams, Practical Exams, Incomplete Exams, Third exams Other.

**The system's core functionalities include:**

- **Efficient Management:** Capabilities to register, delete, shift, and copy exams with accuracy.
- **Database Integration:** Reliable reading from and updating the database to maintain accurate records.
- **Student distribution:** random distribution of students across available labs, ensuring optimal utilization within allowed limits.
- **Platform Compatibility**: Full support for a wide range of exam platforms, including A+, OE, JUSTLMS, MAHARAT, OES, HEAC, and FREE**.**

**Key system attributes:**

- **Performance:** The system must operate with speed and efficiency to handle large volumes of data.
- **User Experience:** A streamlined and intuitive user interface ensures ease of use and minimizes operational time.
- **Security:** Strong measures to protect data integrity and confidentiality.
- **Scalability**: accepting variable workloads and institutional growth**.**

- **Maintainability**: Designed for ease of updates and troubleshooting**.**
- **Usability:** user friendly interface.
- **and Availability:** Ensures reliability and accessibility for all stakeholders.

This system aims to deliver a robust, seamless, and secure solution for managing academic exams.

## 1.2 Project Motivation

**Reasons for Developing This Project?**

**The decision to develop this project stems from the limitations and inefficiencies of the current system. Key challenges include:**

1. **Inefficient Exam Registration:**
   - The current system consumes excessive human and computational resources, making the exam registration process time-consuming and cumbersome.
   - Efficiency in operations, including faster processing times, is urgently needed.
2. **Complex and Overcrowded Interface:**
   - The existing system features a cluttered interface, which complicates and exhausts users during the exam creation process.
3. **Lack of Scalability and Maintainability:**
   - The current system is difficult to adapt to new functionalities.
   - It has reached a level of obsolescence where rebuilding a new system is more practical than attempting further modifications.

**Importance of the Project?**

**This project aims to address and overcome these limitations by delivering:**

- **Scalability:** A system designed to grow with institutional needs and handle increasing workloads.
- **Maintainability:** Simplified processes for adding new features and ensuring system longevity.
- **Improved Usability:** A user-friendly interface that streamlines operations, reduces effort, and enhances productivity.

**Innovative Features Introduced by This Project?**

**The project introduces several new ideas to enhance functionality and user experience, including:**

1. **Lab Availability Checker:**
   - The system will verify lab availability before assigning them for exams.
2. **Enhanced User Interface:**
   - A cleaner, more intuitive design that reduces complexity and improves ease of use.
3. **Quick Add Functionality:**
   - New semester-related details can be added swiftly using the "Quick Add" feature, significantly speeding up administrative tasks.

This project is designed to address the pain points of the current system while introducing innovative features for efficiency and ease of use.

## 1.3 Problem Statement

The existing system suffers from several critical shortcomings that hinder its effectiveness and usability:

1. **Performance Issues**:
   - The current system is outdated, slow, and inefficient, resulting in significant delays in operations.
   - 
2. **Scalability and Maintainability**:
   - It lacks scalability, making it incapable of accommodating the increasing number of students and their future growth.
   - Adding new functionalities is complex and time-consuming, reducing its maintainability.
3. **Crowded User Interface (not user friendly)**:
   - The interface is cluttered and unintuitive, leading to confusion and inefficiency during tasks.
4. **Time-Consuming Processes**:
   - The system demands excessive time for even basic operations, negatively impacting productivity.

5-**Old technology.**

**Proposed Solution**

The new system is designed to address these challenges by:

- **Improved Efficiency**:
  - Ensuring faster operations and reducing the time required for tasks, thereby enhancing user productivity.
- **Scalability**:

  - Simplifying the process of adding new functionalities when needed.
  - making the system modular by separation of concerns.
- **User-Friendly Interface**:
  - Featuring a clear, intuitive, and fluid user interface where every function is distinct and easy to navigate.

This new system aims to provide a robust, scalable, and user-centric solution that resolves the limitations of the current system and sets a foundation for future growth and enhancements.

## 1.4 Project Aim and Objectives

**Project Aim**

The primary aim of this project is to develop a modern, efficient, and scalable exam management system that overcomes the limitations of the current system. The new system will streamline the exam registration process, enhance user experience, and provide a robust foundation for future scalability and functionality enhancements.

**Project Objectives**

By the end of this project, the following objectives are expected to be achieved:

1. **Efficiency and Performance**:
   - Design and implement a system that significantly reduces the time required for exam registration and related operations.
   - Ensure faster data processing and minimize human intervention for routine tasks.

2. **Scalability and Maintainability**:
   - Develop a maintainable framework that allows easy integration of new functionalities over time.
3. **User-Friendly Interface**:
   - Create an intuitive, clear, and fluid user interface to make operations straightforward and minimize user effort.
4. **Resource Optimization**:
   - Introduce functionality to check lab availability before allocation to ensure optimal use of resources.
   - Incorporate features that efficiently allocate nearby labs for the same professor, improving logistical coordination.
5. **Productivity Enhancement**:
   - Reduce operational bottlenecks by introducing a "Quick Add" feature for faster semester-related updates.
   - Streamline the workflow to increase user productivity and reduce administrative workload.

# CHAPTER 2: PLANNING PHASE

## 2.1 Scope of the project

The **scope of the project** defines the objectives and boundaries of the **Exam Registration System**, detailing what the system is set to accomplish, the key features, and the stakeholders involved. The system aims to facilitate the registration of students for exams, manage exam schedules, and allocate computer labs for exam sessions.

**System Boundaries**

**Internal Boundaries**:

- **Institutional Environment**: The system operates within the confines of the university, specifically within the **E-Learning Center** and the **Information Technology & Communications Center**. These departments will manage and maintain the system.

**External Boundaries**:

- **Outside the Institution**: The system will not allow external access beyond the university network to ensure security and privacy.

**Key Stakeholders and Their Roles**:

- **Admins**: Have full control over the system, including functionalities such as exam registration, deletion, shifting, and scheduling.
- **Faculty Members**: Can view the exam dates and schedules but cannot modify the system's core functionalities.

**Environment for System Boundary**:

- **Admin Control**: Full functionality for system administrators to manage the entire registration process.

- **Faculty Access**: Limited to viewing exam schedules.

**Specified Features and Functions**

The **Exam Registration System** will include the following core features and functions:

**Functions:**

**Exam Registration**, **Exam Deletion, Exam Shift, Exam Scheduling**

**Features:**

- **User-Friendly Interface**
- **Nearby labs**: Allocate students to nearby labs for exams scheduled in the same session by the same professor.
- **Lab Availability Checker**: Ensures labs are only allocated when available. If a lab is full, students will be assigned to a different lab.

**-----------------------------------------------------------------------------------------**

**Stakeholders**

**Primary Stakeholders:**

- **Customers/Clients**: **Students** will be the end users, accessing the system to view their exam sessions and computer lab assignments.
- **Employees**: **System admins** and any authorized personnel responsible for managing exam registration and system functionality.
- **Suppliers/Contractors**: The **institution** (Jordan University of Science and Technology) will provide the necessary infrastructure, including servers, domains, portals, computers, and financial support.

**Secondary Stakeholders:**

- **Regulatory Authorities**: The **Jordanian Government** will ensure that the system complies with national regulations and operates legally.

---

### Responsibilities of Team Members

- **Sulaf and Omar**: Both team members share equal responsibilities in the design, development, and deployment of the Exam Registration System. They will work collaboratively to ensure all functionalities are properly implemented and tested.

---

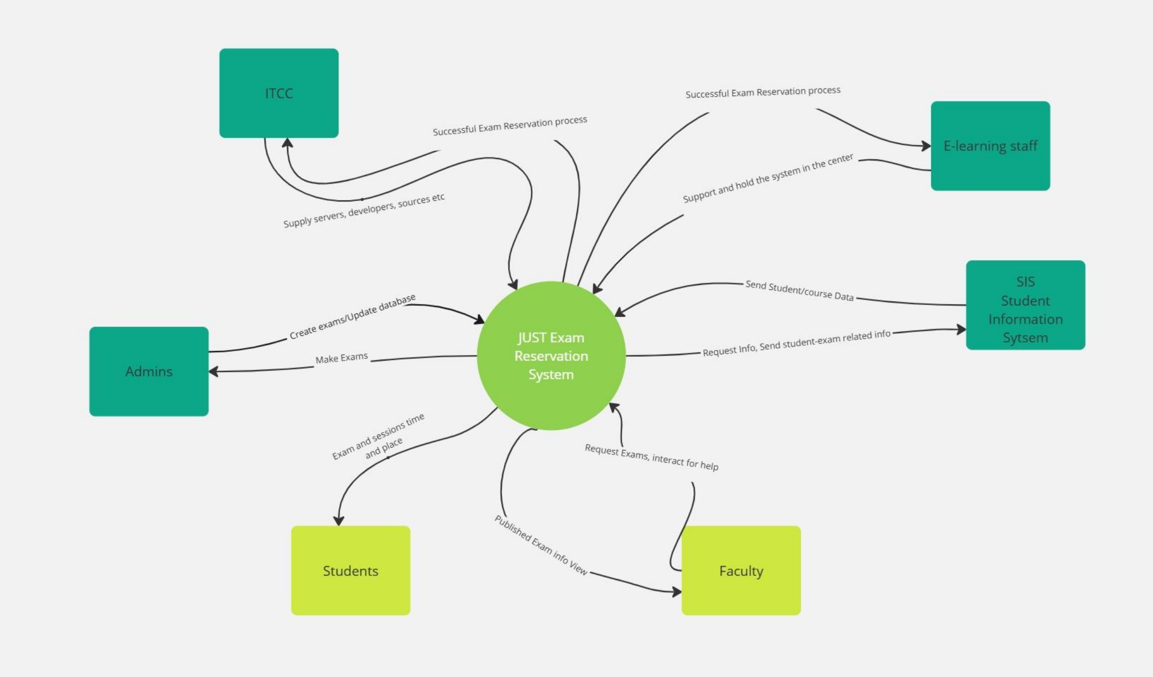
### Verification and Approval Procedures

To verify and approve the system's functionality, the following steps will be taken:

- **System Testing**: After development, the system will undergo frequent testing by **E-Learning Center** staff and initial users to evaluate its functionality and usability.

---

### Context diagram.


ITCC
Successful Exam Reservation process
Supply servers, developers, sources etc
Successful Exam Reservation process
E-learning staff
Support and hold the system in the center
Send Student/course Data
SIS
Student
Information
Sytsem
Request Info, Send student-exam related info
Create exams/Update database
Make Exams
Admins
JUST Exam
Reservation
System
Exam and sessions time
and place
Students
Request Exams, interact for help
Published Exam info View
Faculty

## 2.2 Project risks and Product risks

**1-Improper Handling of Database Queries and Procedures:**

There is a risk of database queries being mishandled or incorrectly implemented, which can lead to data corruption, loss, or inconsistent system behavior. Ensuring robust query handling and validation is critical.

**Mitigation:**

1-Implement thorough query validation and optimization techniques.

2-Conduct regular code reviews to catch errors early.

3-Perform testing.

**2-Data Handling Risks:**

Improper insertion, alteration, or deletion of data in the database could lead to inaccurate or inconsistent records. Strict validation checks and data integrity controls must be in place to mitigate this risk.

**Mitigation:**

1-Implement strict data validation checks at all points of data entry.

2-Enforce data integrity controls such as primary and foreign key constraints.

**3-Security Risks:**

The risk of sensitive information, such as passwords and emails, being leaked or compromised. This can expose the system to data breaches. Strong encryption, secure authentication mechanisms, and regular security audits are necessary to mitigate these risks.

**Mitigation:**

1-the institution will make sure to keep track of security.

**4-Unhandled Cases (Control Flow Risks):**

Unexpected behavior may occur due to unhandled cases or control flows in the system, such as accidental deletion of exams or insertion of students into a full lab. Comprehensive error handling and thorough testing are required to address edge cases.

**Mitigation:**

1-Implement robust errors in handling for all potential control flows.

**5-Usability Risks:**

A poorly designed user interface (UI) can cause confusion, particularly when buttons with similar appearances perform different actions (e.g., a "Delete" button and an

"X" button that could be mistaken for a page exit). Clear labeling, tooltips, and UI design best practices should be applied to prevent usability issues.

**Mitigation:**

1-Use clear and consistent labeling for buttons, including tooltips and visual cues.

2-Conduct usability testing with real users to gather feedback and iterate on the design.

**6-Late Project Delivery:**

Delays in the development process could lead to the late delivery of the project. Setting clear milestones, regular progress reviews, and addressing potential roadblocks early are essential to meet deadlines.

**Mitigation:**

1-Set clear, realistic milestones with regular progress reviews.

3-Proactively address any roadblocks or risks that could lead to delays by having contingency plans in place.

## 2.3 Feasibility study

The proposed project does not require additional financial investment, as it will serve as a replacement for the current system. Consequently, the new system will leverage the existing infrastructure, including servers, databases, hardware, and user access, ensuring no additional costs for these resources.

Furthermore, the new system will be developed using open-source tools and coding environments, meaning no additional financial aid will be necessary for software procurement. This approach minimizes project costs while utilizing widely adopted, cost-effective development tools.

In terms of technology, the new system will be built with the same core web technologies and programming languages that are already familiar to the current development team and end-users. This ensures compatibility with existing systems and provides a smooth transition, as it aligns with the tools and technologies most used in the organization's current infrastructure.

## 2.4 Project Schedule

The project must be done by June.

## 2.5 Project Software and Hardware Requirements

**Software Requirements**

- **Tools and Environments**:
    1. **Laravel Composer**: Dependency management for PHP projects.
    2. **PHP**: Programming language for the web.
    3. **Tailwind CSS**: Utility-first CSS framework for designing custom user interfaces.

**Integrated Development Environment (IDE)**:

- **Visual Studio Code**: A lightweight, powerful code editor with support for various programming languages and extensions.
- **PHP storm**.

- **Database Tools**:

1. **Toad for Data Generation**: Tool for generating and managing test data in databases.
2. **Oracle Express**: Free version of Oracle Database for development and testing purposes.
3. **SQL Developer**: Oracle tool for database development, management, and SQL query execution.

- **Servers**:
    - **XAMPP**: Open-source, cross-platform web server solution package, which includes Apache, MySQL, PHP, and Perl.

**Hardware Requirements**

Gaming laptops with high computing abilities.

**Ram:**8 GB

**CPU**: Core i5 (10th generation and 12th generation).

# CHAPTER 3: RELATED EXISTING SYSTEM

**Existing System Overview:**

The current system in use is the university's exam registration platform, which allows Admins to register for exams. This system provides the basic functionality necessary to facilitate exam registration; however, it is somewhat limited in terms of user experience and system performance. While it is functional and serves its purpose, there are notable areas for improvement, particularly in terms of efficiency, user interface, and the ability to scale with evolving needs.

**Proposed System: Key Enhancements:**

Our proposed system closely mirrors the functionality of the existing system, with the core feature of exam registration remaining central to both platforms. However, the new system is designed to be more efficient, faster, and optimized for better performance. Additionally, it will incorporate a modernized user interface that improves the overall user experience for students and administrators alike. The system will also offer expanded functionality to meet the growing demands of exam registration at the institution.

**Functional Comparison:**

While both systems share similar functional requirements, our system provides a more detailed and comprehensive solution. It will not only replicate the existing features but also add new capabilities that cater to the evolving needs of the institution. These enhancements aim to streamline the registration process, reduce system load, and minimize the potential for user error.

**Non-Functional Improvements:**

A key distinguishing factor between the current system and our proposed system is the non-functional requirements. While the existing system remains operational, it has certain weaknesses in areas such as performance, scalability, and usability. Our system addresses these shortcomings by focusing on these non-functional aspects, ensuring a smoother, faster, and more reliable experience.

**Conclusion:**

In conclusion, while the existing system is functional and continues to serve its purpose, the proposed system offers significant improvements in terms of performance, user interface, and overall functionality. By building upon the foundation of the existing system and optimizing key areas, the new system will provide a more efficient and user-friendly solution for exam registration at the university.

# CHAPTER 4: REQUIREMENT ENGINEERING AND ANALYSIS

## 4.1 Used Techniques

To gather the necessary requirements for the project, we utilized a combination of **direct stakeholder interviews** and **system observation**.

1. **Stakeholder-Interviews**:

   We visited the **eLearning Center** and conducted discussions with the **system administrators** to collect detailed insights about the existing system and their specific requirements for the new system. These interactions helped to identify both functional and non-functional requirements, as well as pain points in the current system.

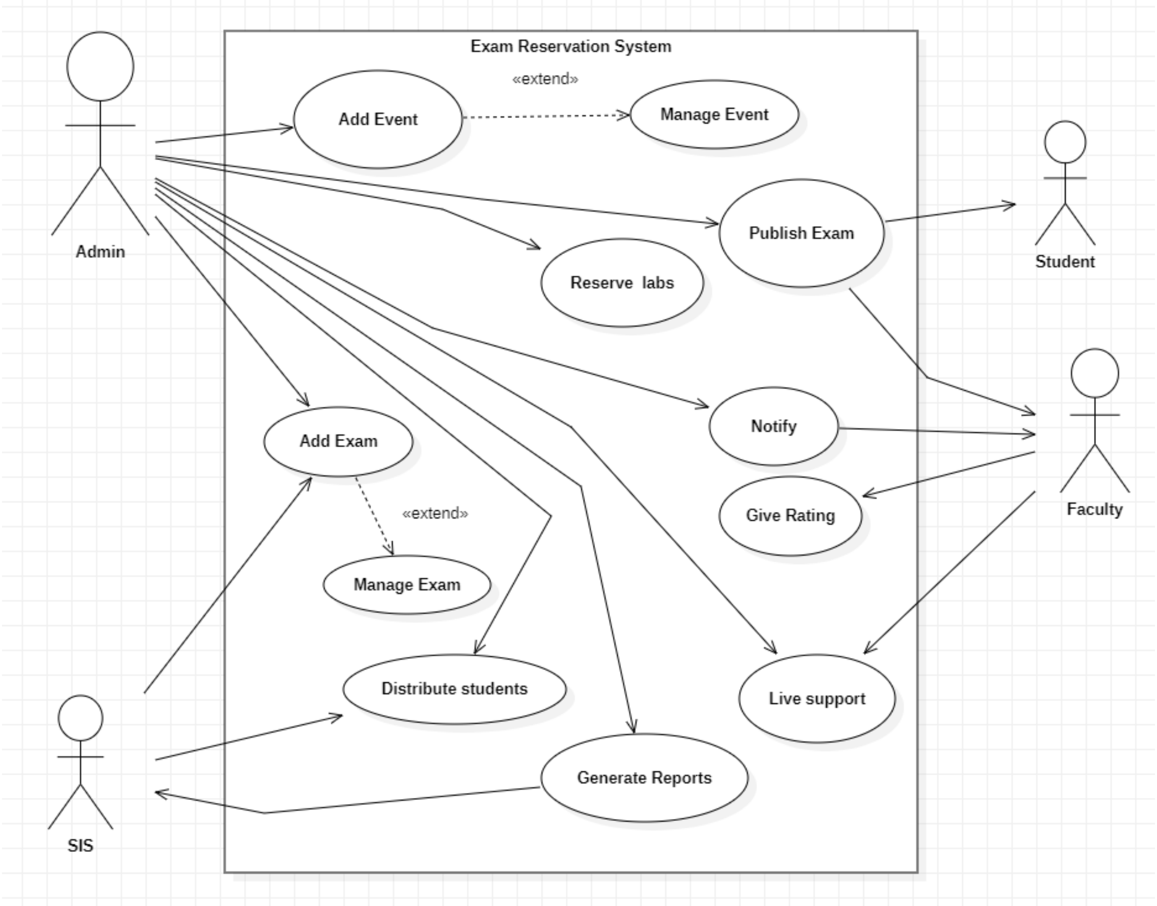
2. **System-Observation**:

   In addition to the interviews, we analyzed the **old system** to understand its structure and functionality. This observation helped identify key areas to replicate and highlighted potential enhancements based on feedback from the system administrators and users.

Through this dual approach of engaging with stakeholders and analyzing the existing system, we were able to accurately document the requirements and expectations for the new system.

## 4.3 Functional Requirement & Modelling

### 4.3.1 Use Case Diagram



### 4.3.2 Use Cases: Description & Details

**Admin use cases:**

**Add Exam:** use case to create exam, choose exam type (first, second. etc), choose exam, choose exam platform (A+, OE. etc), choose semester.
Includes remove exam, copy exam, shift exam, hide exam, schedule exam.

**Add Event:** to create Extracurricular exams, similar functionality to add exam.

**Publish Exam:** to show the exam on portal and make it accessible to be viewed.

**Reserve labs:** to reserve a lab for the added exam.

**Notify:** to send notifications with faculty about the exams, schedules, labs etc. **(shared use case with faulty)**

**Generate Reports:** generate reports based on the feedback that comes back from the faculty, as well as any flaws or defects in the system or the reservation process.

**Distribute Students:** to distribute students over the reserved labs for the corresponding exam.

**Faculty use cases:**

**View Exam Schedule:** see published exams on the portal (based on publish use case). **(shared use case with students)**

**Send Feedback:** to send feedback based on the interaction with the admin during the system.

**Request Exam:** to request a specific exam from the system.

**SIS system use cases:**
**Send student info:** send student's info to the reservation system.

## 4.4 Nonfunctional Requirements: Quality & Constriants

| NFR Category | Description |
|---|---|
| Security | Only admins and authorized users can access and execute operations via the system, any deletion button be made harder to choose in case it alters the database permanently |

| | |
|---|---|
| Reliability | The system should hold on to any last operation made via the system before any crash, when system is up again, no changes will be lost |
| Performance | With Laravel framework and code that conforms to modern rules, performance should be relatively fast |
| Maintainability | With code that depends on separation and abstraction of components, the system will be maintainable |
| Scalability | If the system is maintainable then it is scalable, with modularization and separation of different code components (separation of concerns), the system will be scalable for future requirements |
| Usability | Friendly, fluid and clear interface would make the system gain very high grade on the usability scale |

# CHAPTER 5: ARCHITECTURE & DESIGN

## 5.1 Software Architecture

Our System is an MVC based architecture, were we have a central database that we fetch information from via models for every class or when needed, and a controller that calls the appropriate methods to provide the view with what it needs to show via the GUI.

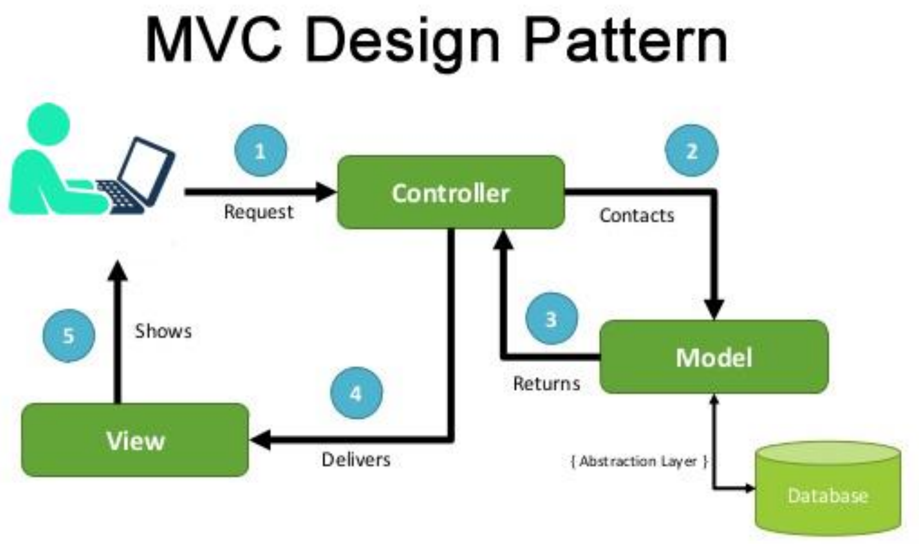
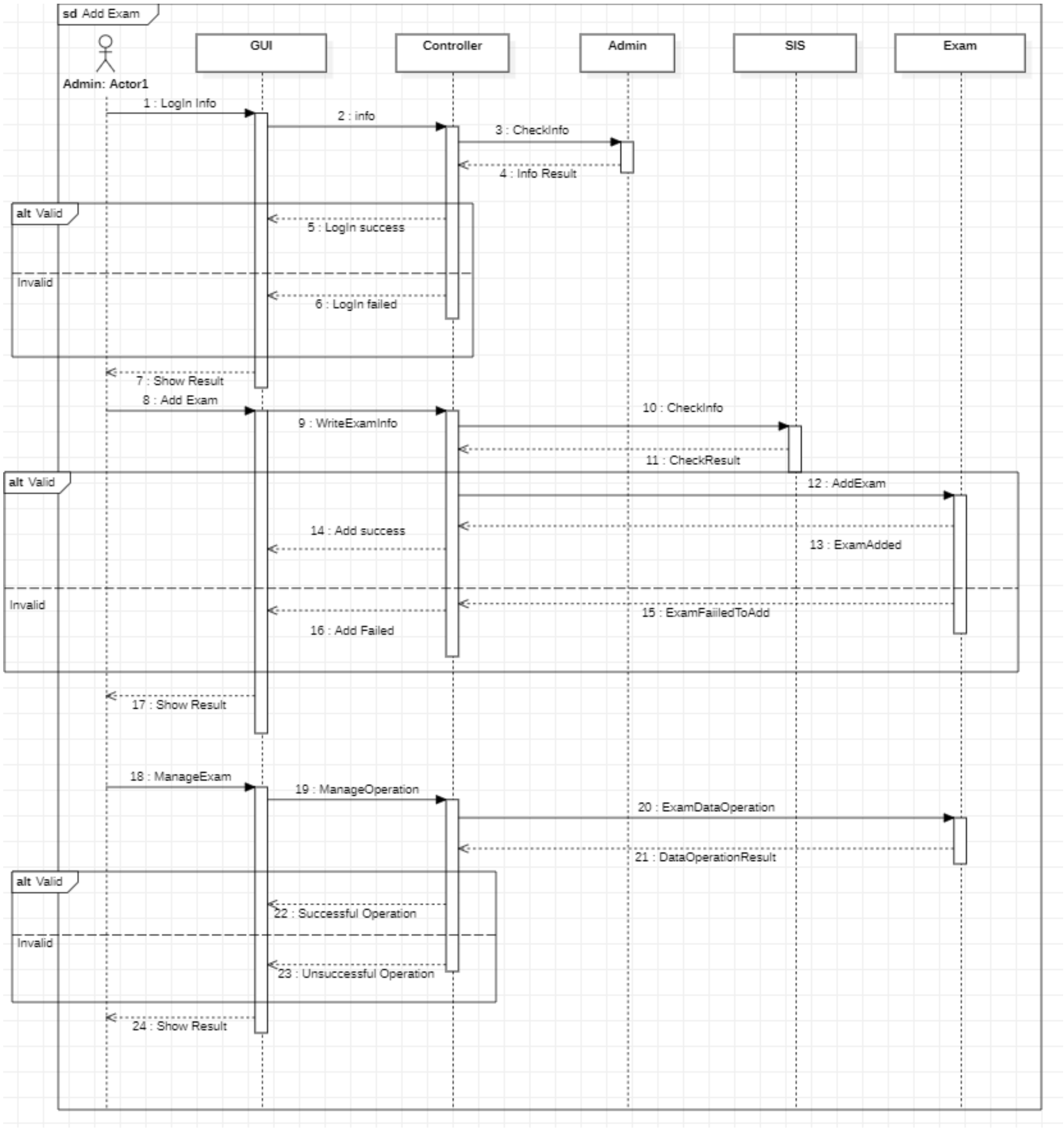


**Figure 1: System Overview**

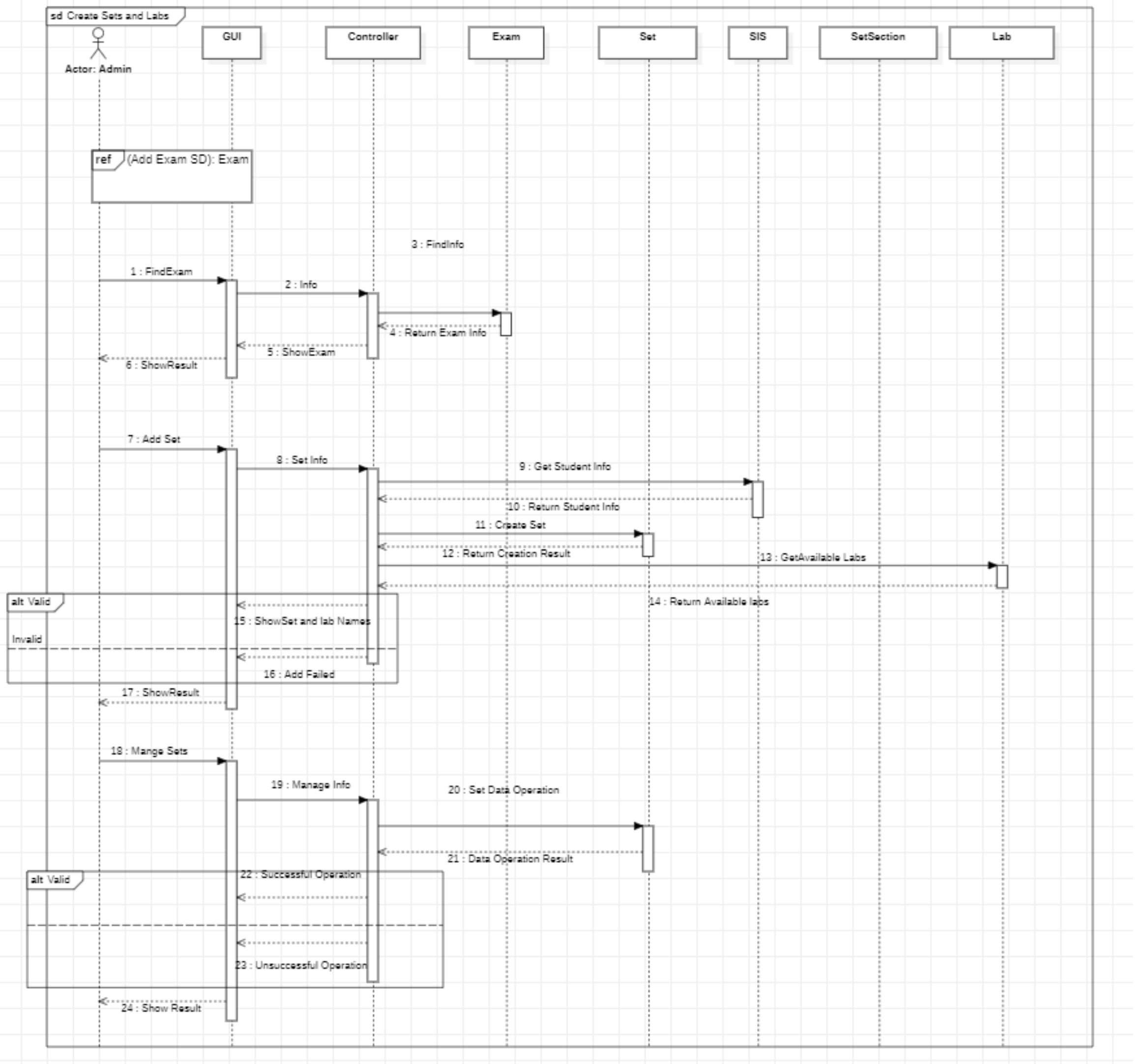
## 5.2 Software Detailed Design

### 5.2.1 Use Cases Internal Interactions

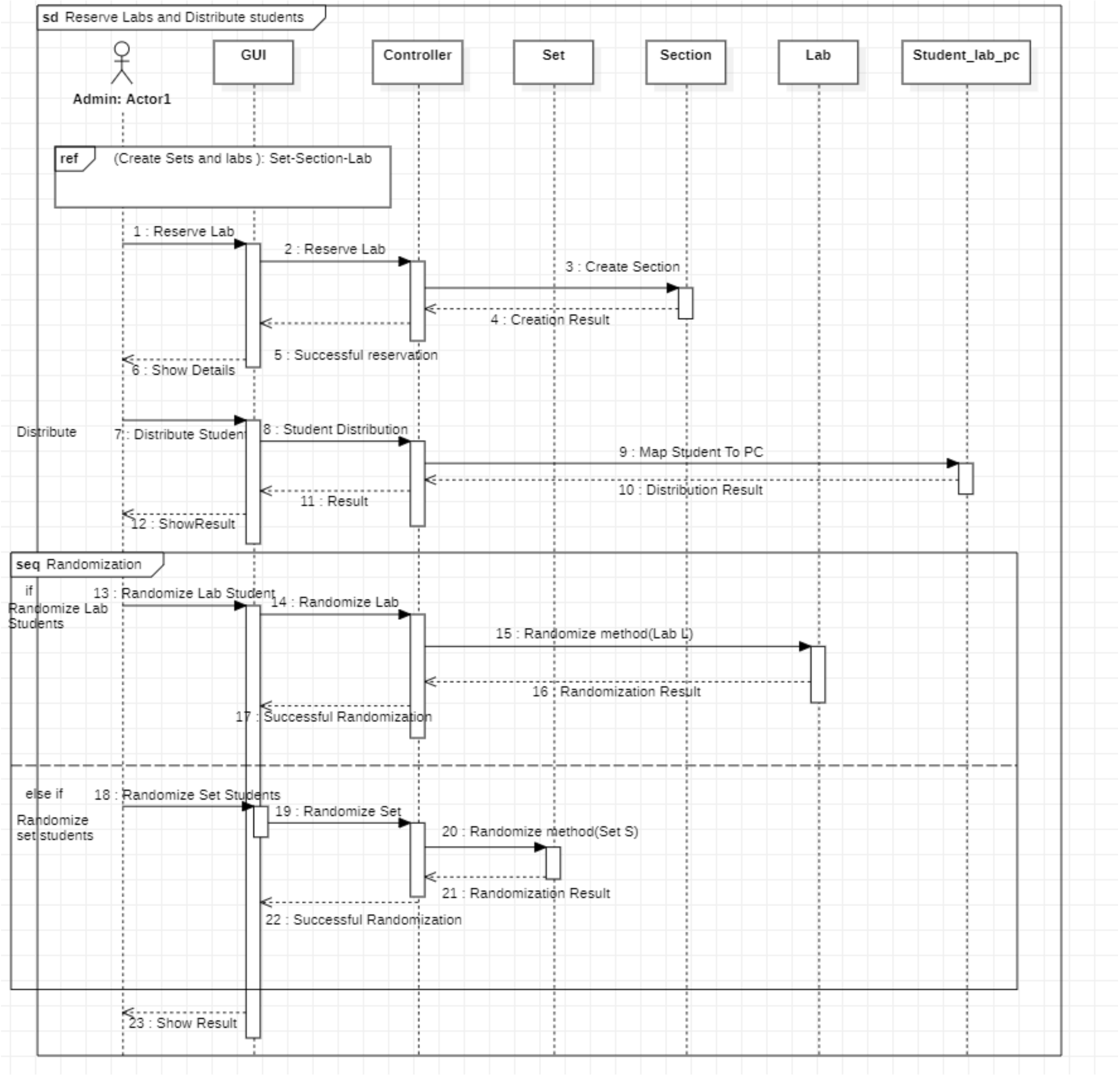
Add Exam SD


sd Add Exam
Admin: Actor1
GUI
Controller
Admin
SIS
Exam
1 : LogIn Info
2 : info
3 : CheckInfo
4 : Info Result
alt Valid
5 : LogIn success
Invalid
6 : LogIn failed
7 : Show Result
8 : Add Exam
9 : WriteExamInfo
10 : CheckInfo
11 : CheckResult
alt Valid
12 : AddExam
14 : Add success
13 : ExamAdded
Invalid
15 : ExamFailledToAdd
16 : Add Failed
17 : Show Result
18 : ManageExam
19 : ManageOperation
20 : ExamDataOperation
21 : DataOperationResult
alt Valid
22 : Successful Operation
Invalid
23 : Unsuccessful Operation
24 : Show Result

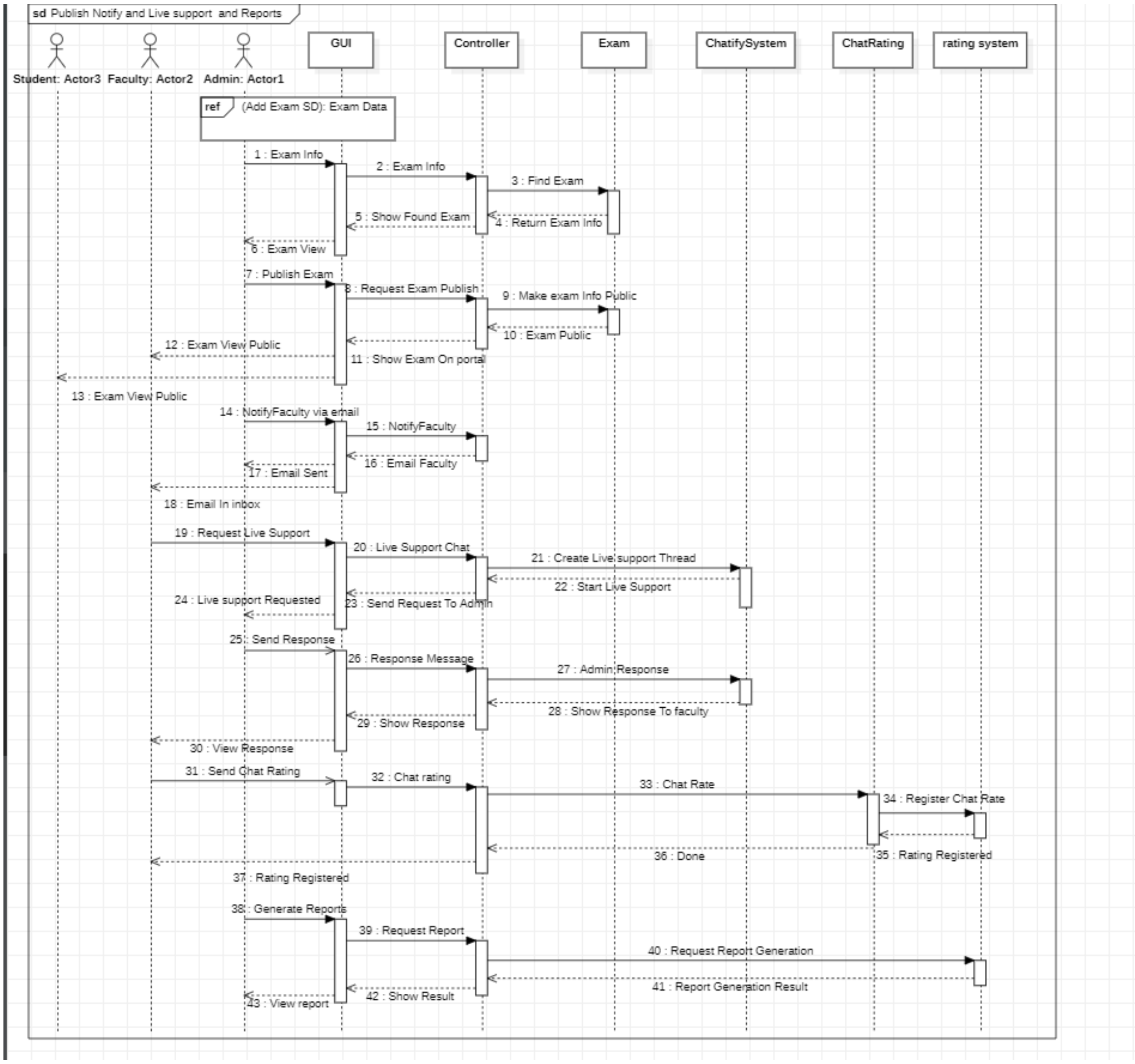
## Sets SD


sd Create Sets and Labs
Actor: Admin
GUI
Controller
Exam
Set
SIS
SetSection
Lab
ref
(Add Exam SD): Exam
3 : FindInfo
1 : FindExam
2 : Info
4 : Return Exam Info
5 : ShowExam
6 : ShowResult
7 : Add Set
8 : Set Info
9 : Get Student Info
10 : Return Student Info
11 : Create Set
12 : Return Creation Result
13 : GetAvailable Labs
alt Valid
14 : Return Available labs
15 : ShowSet and lab Names
Invalid
16 : Add Failed
17 : ShowResult
18 : Mange Sets
19 : Manage Info
20 : Set Data Operation
21 : Data Operation Result
alt Valid
22 : Successful Operation
23 : Unsuccessful Operation
24 : Show Result

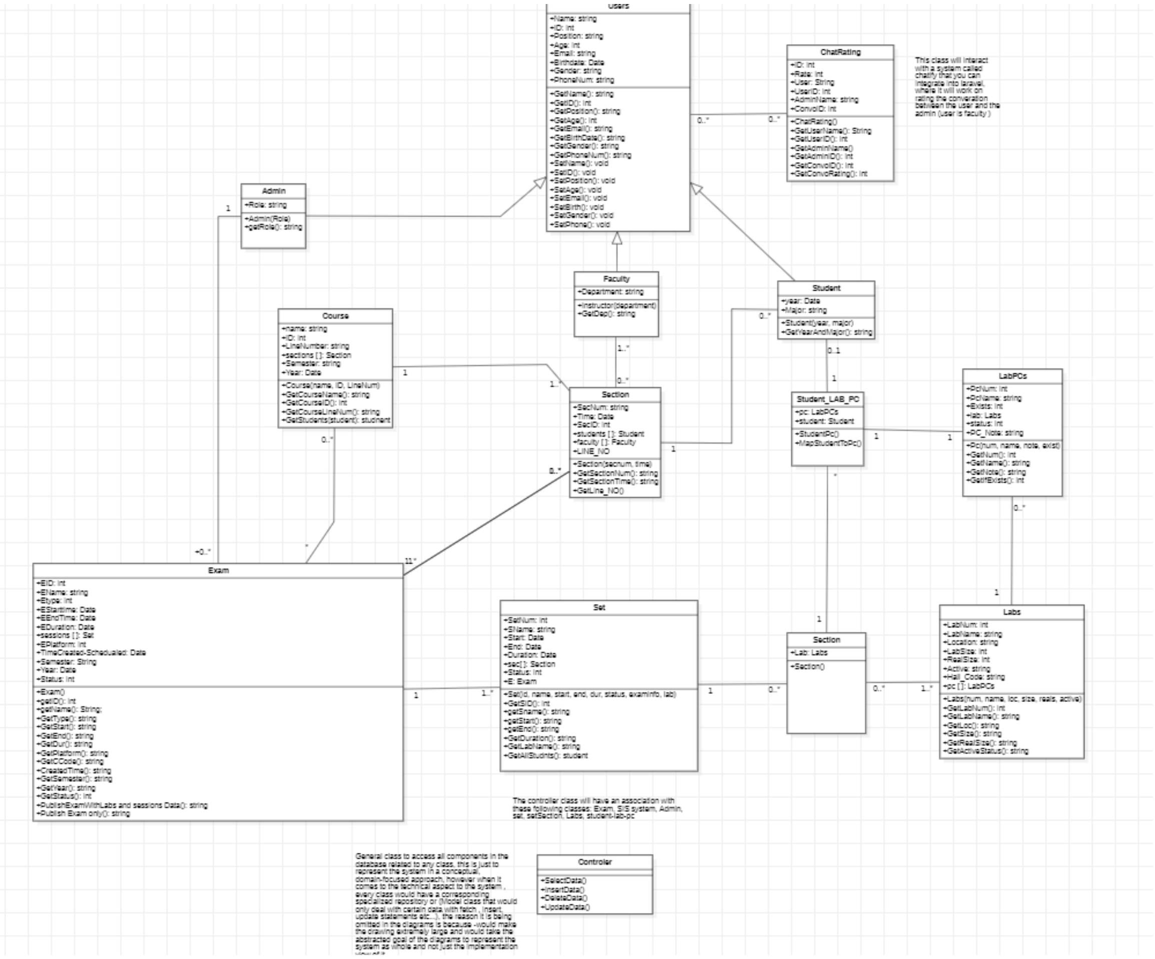
Reserve Labs and Distribute Students SD


sd Reserve Labs and Distribute students
Admin: Actor1
GUI
Controller
Set
Section
Lab
Student_lab_pc
ref (Create Sets and labs ): Set-Section-Lab
1 : Reserve Lab
2 : Reserve Lab
3 : Create Section
4 : Creation Result
5 : Successful reservation
6 : Show Details
Distribute
7 : Distribute Student
8 : Student Distribution
9 : Map Student To PC
10 : Distribution Result
11 : Result
12 : ShowResult
seq Randomization
if Randomize Lab Students
13 : Randomize Lab Student
14 : Randomize Lab
15 : Randomize method(Lab L)
16 : Randomization Result
17 : Successful Randomization
else if Randomize set students
18 : Randomize Set Students
19 : Randomize Set
20 : Randomize method(Set S)
21 : Randomization Result
22 : Successful Randomization
23 : Show Result

Publish Exam, Notify Faculty, Live Support, Generate Reports SD



### 5.2.2 Class Diagram

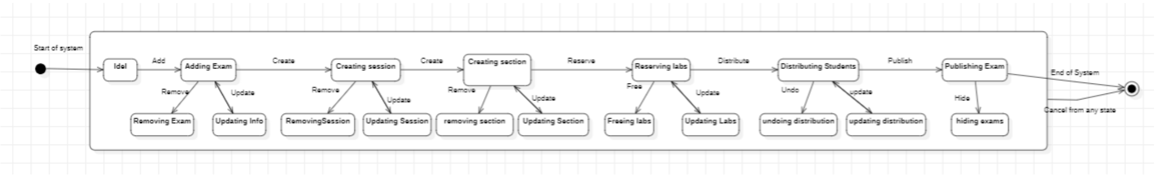


This class will interact with a system called chatify that you can integrate into laravel, where it will work on rating the converation between the user and the admin (user is faculty )

The controller class will have an association with these following classes: Exam, SIS system, Admin, set, setSection, Labs, student-lab-pc

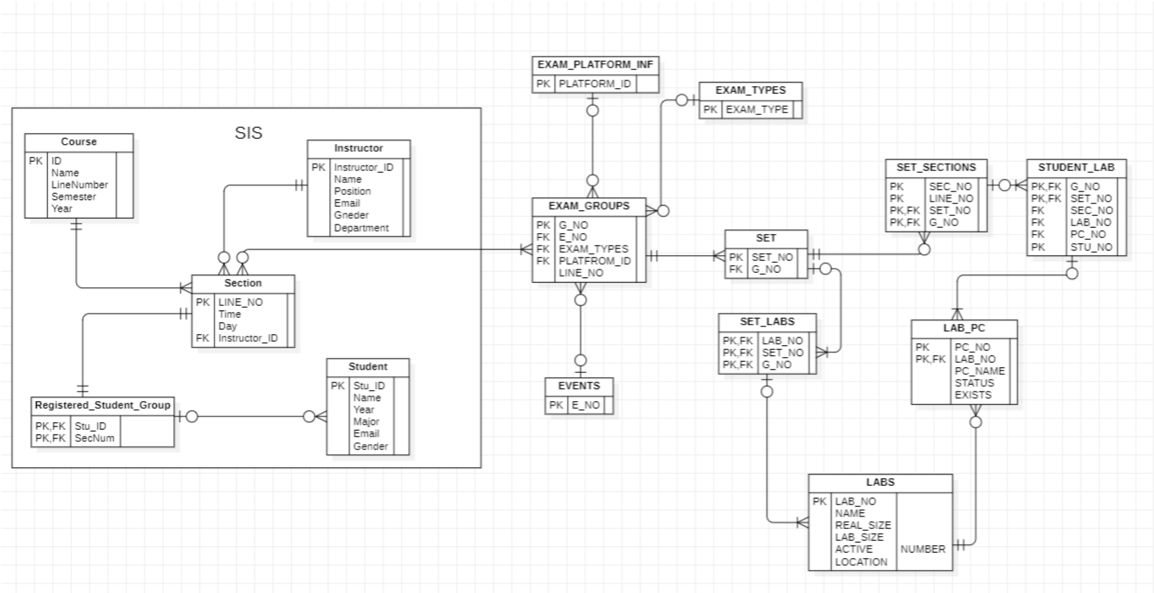
General class to access all components in the database related to any class, this is just to represent the system in a conceptual, domain-focused approach, however when it comes to the technical aspect to the system , every class would have a corresponding specialized repository or (Model class that would only deal with certain data with fetch , insert, update statements etc...), the reason it is being omitted in the diagrams is because -would make the drawing extremely large and would take the abstracted goal of the diagrams to represent the system as whole and not just the implementation view of it.

### 5.2.3 State transition Diagram

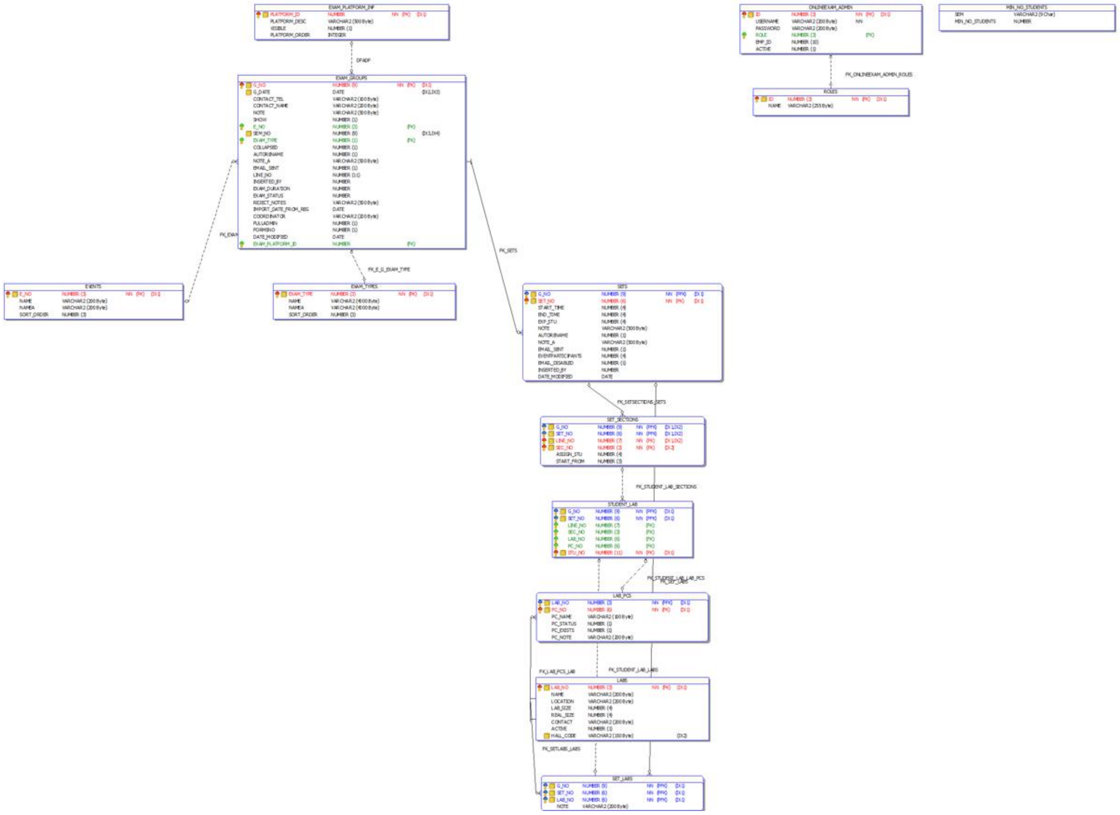


### 5.2.4 Data Storage Organization

Our Cut-Down Data Base



# The Real Data base



# CHAPTER 6: IMPLEMENTATION PLAN & PROTOTYPING

## 6.1 Introduction

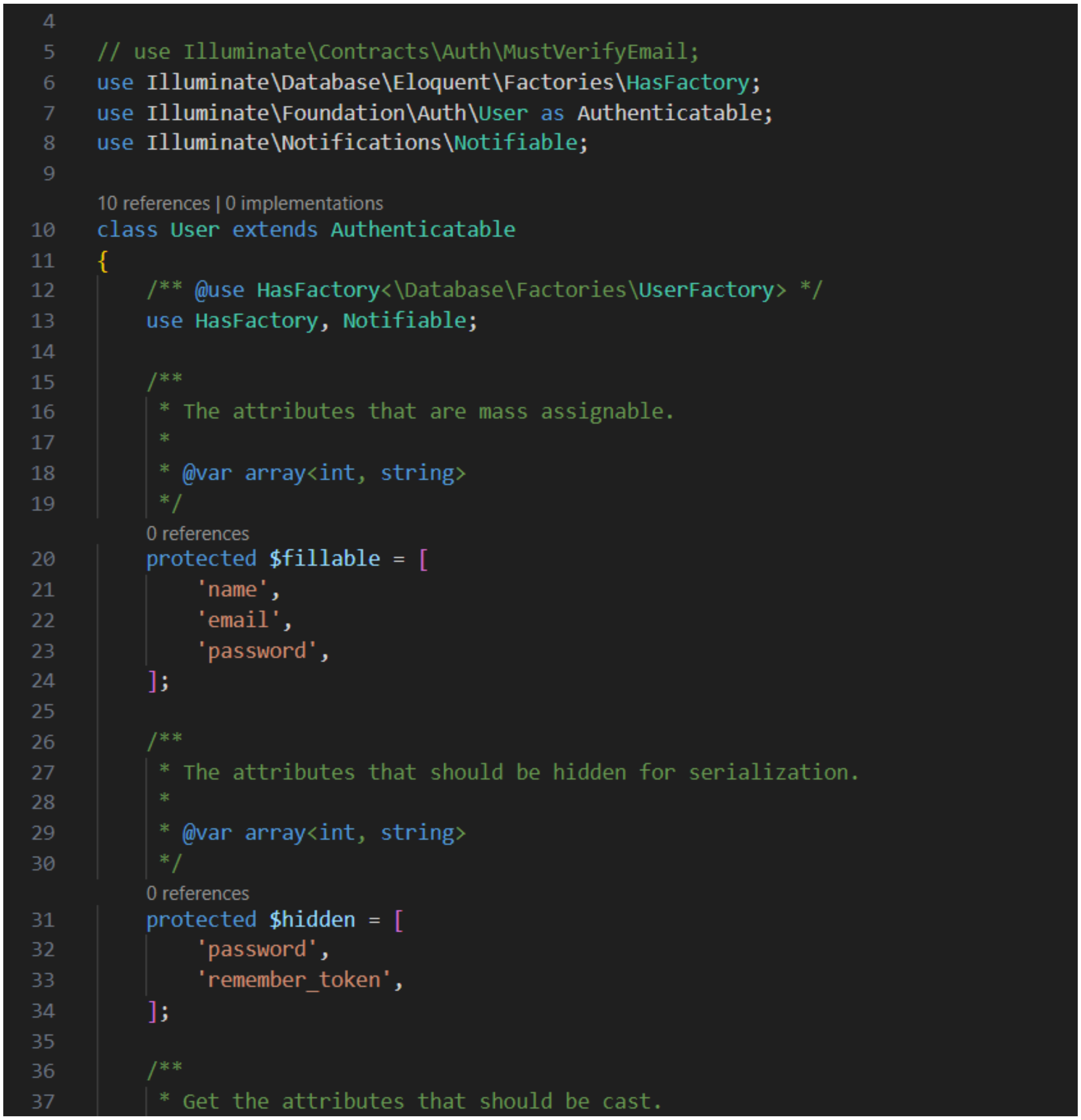
The implementation relies on modern technologies to ensure reliability, security, and efficiency. The key elements of the implementation include:

- **Database**: Oracle Express, utilized for its performance and scalability.
- **Programming Language**: PHP, chosen for its compatibility with web applications and support for Laravel, the framework used.
- **Framework**: Laravel, used for its elegant syntax, robust feature set, and strong community support.
- **Frontend Design**: Tailwind CSS for styling, ensuring a responsive and user-friendly interface.
- **Development Tools**: Visual Studio Code for coding, XAMPP for local server setup, and Toad for Data Generation for database testing.
- **Third-Party Tools**: Integration of additional software components to facilitate specific functionalities like email communication and report generation, chat system.

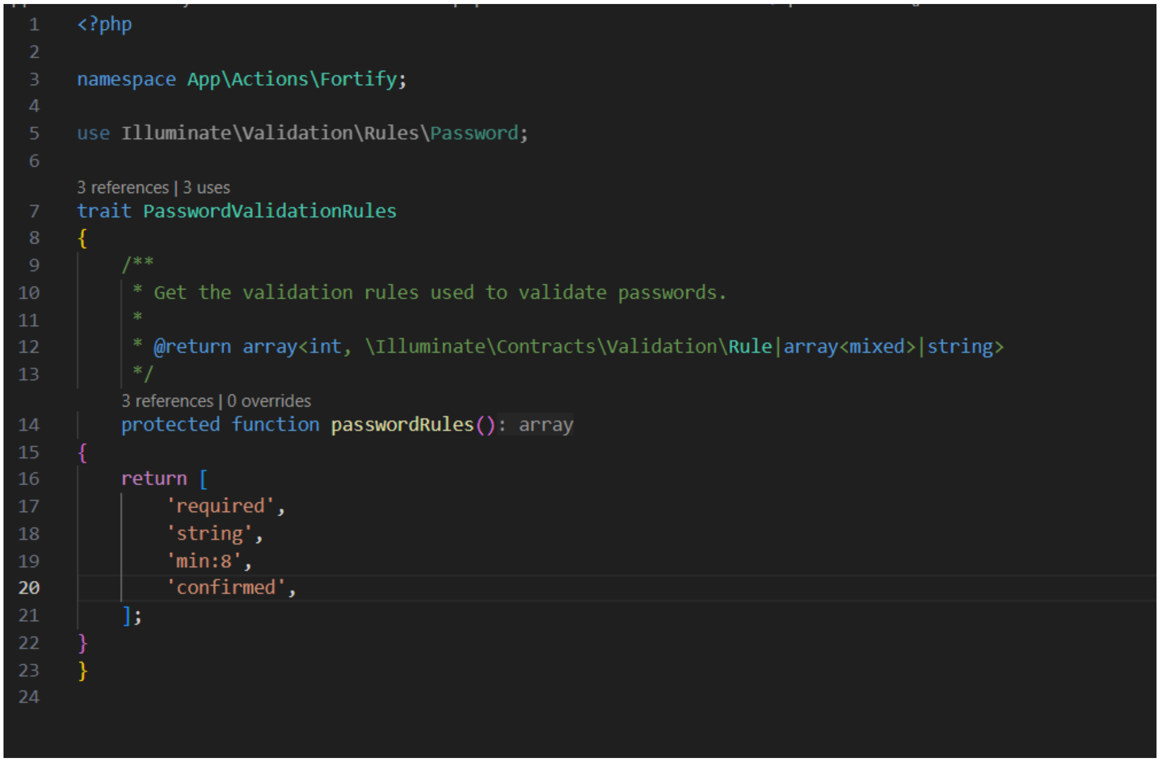
## 6.2 Prototyping

1-We Implemented the Log-In page of the current system, which was done by Laravel PHP blade system using Fortify library (for authentication) which encrypts passwords in the data base for protection.

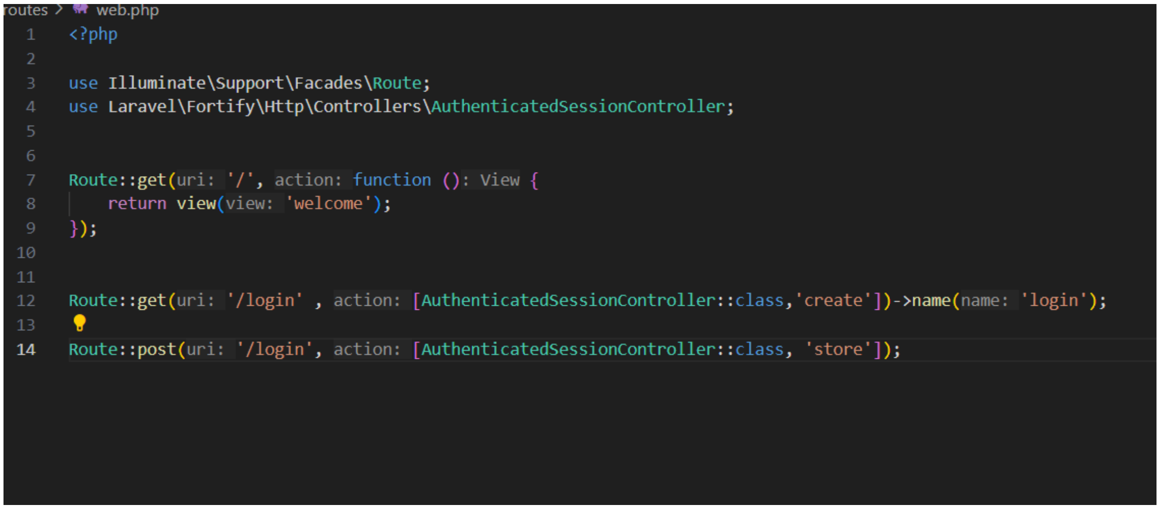
2-We decided on the color schema for the website.



```php

// use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;

10 references | 0 implementations
class User extends Authenticatable
{
    /** @use HasFactory<\Database\Factories\UserFactory> */
    use HasFactory, Notifiable;

    /**
     * The attributes that are mass assignable.
     *
     * @var array<int, string>
     */
    0 references
    protected $fillable = [
        'name',
        'email',
        'password',
    ];

    /**
     * The attributes that should be hidden for serialization.
     *
     * @var array<int, string>
     */
    0 references
    protected $hidden = [
        'password',
        'remember_token',
    ];

    /**
     * Get the attributes that should be cast.
```

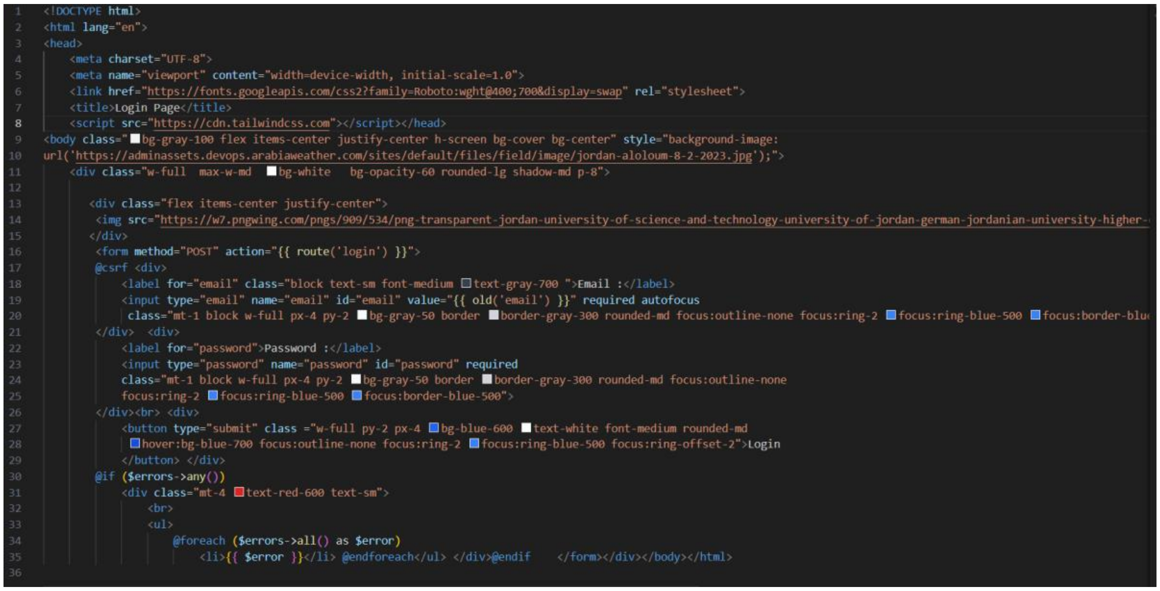


```php
<?php

namespace App\Actions\Fortify;

use Illuminate\Validation\Rules\Password;

3 references | 3 uses
trait PasswordValidationRules
{
    /**
     * Get the validation rules used to validate passwords.
     *
     * @return array<int, \Illuminate\Contracts\Validation\Rule|array<mixed>|string>
     */
    3 references | 0 overrides
    protected function passwordRules(): array
{
    return [
        'required',
        'string',
        'min:8',
        'confirmed',
    ];
}
}
```

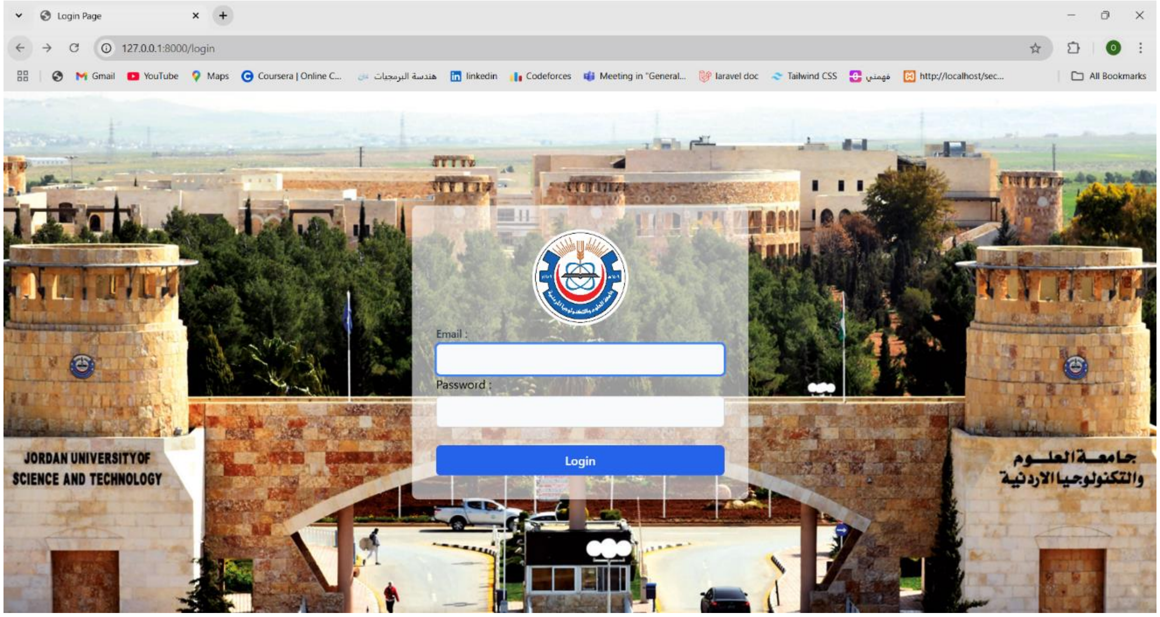


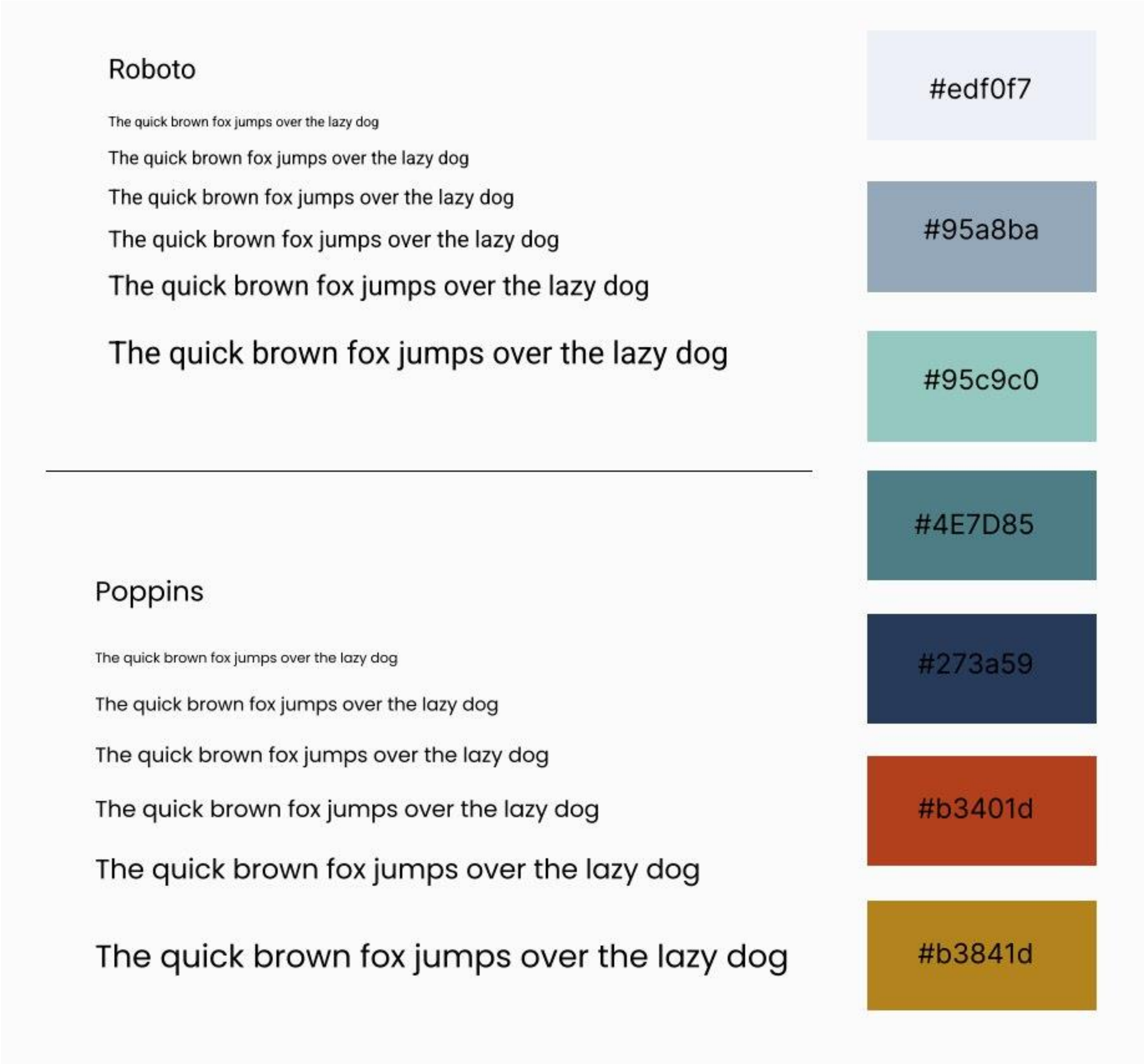
outes > web.php

```php
<?php

use Illuminate\Support\Facades\Route;
use Laravel\Fortify\Http\Controllers\AuthenticatedSessionController;


Route::get(uri: '/', action: function (): View {
    return view(view: 'welcome');
});


Route::get(uri: '/login' , action: [AuthenticatedSessionController::class,'create'])->name(name: 'login');

Route::post(uri: '/login', action: [AuthenticatedSessionController::class, 'store']);
```



```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href="https://fonts.googleapis.com/css2?family=Roboto:wght@400;700&display=swap" rel="stylesheet">
    <title>Login Page</title>
    <script src="https://cdn.tailwindcss.com"></script></head>
<body class="bg-gray-100 flex items-center justify-center h-screen bg-cover bg-center" style="background-image:
url('https://adminassets.devops.arabiaweather.com/sites/default/files/field/image/jordan-aloloum-8-2-2023.jpg');">
    <div class="w-full  max-w-md  bg-white   bg-opacity-60 rounded-lg shadow-md p-8">

       <div class="flex items-center justify-center">
        <img src="https://w7.pngwing.com/pngs/909/534/png-transparent-jordan-university-of-science-and-technology-university-of-jordan-german-jordanian-university-higher-
       </div>
        <form method="POST" action="{{ route('login') }}">
        @csrf <div>
            <label for="email" class="block text-sm font-medium text-gray-700 ">Email :</label>
            <input type="email" name="email" id="email" value="{{ old('email') }}" required autofocus
             class="mt-1 block w-full px-4 py-2 bg-gray-50 border border-gray-300 rounded-md focus:outline-none focus:ring-2 focus:ring-blue-500 focus:border-blu
        </div>  <div>
            <label for="password">Password :</label>
            <input type="password" name="password" id="password" required
            class="mt-1 block w-full px-4 py-2 bg-gray-50 border border-gray-300 rounded-md focus:outline-none
            focus:ring-2 focus:ring-blue-500 focus:border-blue-500">
        </div><br> <div>
            <button type="submit" class ="w-full py-2 px-4 bg-blue-600 text-white font-medium rounded-md
             hover:bg-blue-700 focus:outline-none focus:ring-2 focus:ring-blue-500 focus:ring-offset-2">Login
            </button> </div>
        @if ($errors->any())
            <div class="mt-4 text-red-600 text-sm">
                <br>
                <ul>
                    @foreach ($errors->all() as $error)
                        <li>{{ $error }}</li> @endforeach</ul> </div>@endif    </form></div></body></html>
```





Roboto

The quick brown fox jumps over the lazy dog

The quick brown fox jumps over the lazy dog

The quick brown fox jumps over the lazy dog

The quick brown fox jumps over the lazy dog

The quick brown fox jumps over the lazy dog

The quick brown fox jumps over the lazy dog

Poppins

The quick brown fox jumps over the lazy dog

The quick brown fox jumps over the lazy dog

The quick brown fox jumps over the lazy dog

The quick brown fox jumps over the lazy dog

The quick brown fox jumps over the lazy dog

The quick brown fox jumps over the lazy dog

#edf0f7

#95a8ba

#95c9c0

#4E7D85

#273a59

#b3401d

#b3841d

# CHAPTER 6: TESTING PLAN

## 6.1 Tools

- **Selenium**: For automated testing of web interfaces.
- **Postman**: To test and validate API endpoints and ensure correct data exchange between system components.
- **JUnit**: For unit testing individual components of the system to ensure they function as expected.
- **Apache JMeter**: For performance and load testing to evaluate the system's behavior under various conditions.
- **Oracle SQL Developer**: To test database queries and validate data integrity.

## 6.2. System Testing Plan (Black box)

- **Functional Testing**:
  - Verify that all features, such as exam creation, lab reservation, and student distribution, function as intended.
  - Ensure role-based access control works correctly for admins, faculty, and students.
- **Performance Testing**:
  - Test system responsiveness and stability under high user load.
  - Evaluate database performance during simultaneous read/write operations.

- **Usability Testing**:
    - Assess the GUI for clarity, ease of navigation, and user satisfaction.
    - Conduct user surveys to gather feedback on system usability.
- **Security Testing**:
    - Test login mechanisms for vulnerabilities such as SQL injection and brute force attacks.
    - Validate data encryption and secure transmission of sensitive information.
- **Error Handling**:
    - Test the system's response to invalid inputs and unexpected scenarios.
    - Verify error messages are clear and guide users to resolve issues.

# CONCLUSIONS

The "JUST Exam Reservation System" addresses critical shortcomings of the current system, including inefficiencies, scalability issues, and a cumbersome user interface. The project aimed to create a robust, modern system to streamline exam management processes and enhance user satisfaction.

**Summary of Problems Found**

1. **Performance Issues**:
   - The outdated system was slow and inefficient, causing delays in operations.
2. **Scalability and Maintainability**:
   - The old system was unable to accommodate the growing needs of the university.
3. **Complex User Interface**:
   - A cluttered design hindered users’ ability to complete tasks efficiently.

**Proposed Solution**

The system to be developed offers a scalable, maintainable, and user-friendly solution. Key features include:

- Faster lab reservation and student distribution.
- A modernized user interface with improved navigation.
- Role-based access control for enhanced security.
- Advanced reporting and communication tools.

**Benefits of Pursuing the Project**

1. **Efficiency**:
   - Reduced operational bottlenecks and faster processing times.

2. **Scalability**:
   - Modular design allows for easy integration of future functionalities.
3. **Improved User Experience**:
   - Intuitive interfaces minimize user errors and improve satisfaction.
4. **Resource Optimization**:
   - Real-time lab availability checks ensure effective resource allocation.
5. **Enhanced Security**:
   - Robust authentication mechanisms protect sensitive data.

By addressing these issues and implementing innovative features, the "JUST Exam Reservation System" sets a new standard for exam management at the university. It not only resolves current challenges but also positions the institution to adapt to future demands, making it a valuable investment in the university’s operational efficiency and student experience.

# REFERENCES

[1] Babineau W., Barry P., Furness Z., "Automated Testing within the Joint Training confederation (JTC)", Proceedings of the Fall 1998 Simulation Interoperability Workshop, Orlando, FL, USA. September 1998.

[2] Banks C. "Introduction to Modeling and Simulation". Chapter 1 in book “Modeling and Simulation Fundamentals: Theoretical Underpinnings and Practical Domains”. Catherine Banks, John Sokolowski Editors. Wiley. New Jersey, 2010.

[3] Booth D., Haas H., McCabe F., Newcomer E., Champion M., Ferris C., Orchard D. “Web Services Architecture”. 2004. <http://www.w3.org/TR/ws-arch/>. Accessed November 2010.

# APPENDIX-A: MANUAL