Software Design Specification

for

Project Grading System (PGS)

Version 2.0

Prepared by
HAMZAH KOURDIEH
AMJAD ALNEMA
June 20, 2024

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
HAMZAH KOURDIEH AMJAD ALNEMA	06/20/24	Adding parameters to sequence diagram Adding space for pages layout (0.8 inch)	2.0

1. Introduction

1.1 Purpose

this document is to present a detailed description of the PGS. It explains the purpose, features, interfaces of the system, constraints under which it must operate and how the system will react. This document is intended for both the stakeholders and the developers and will be proposed to Dr. Varol for his approval.

1.2 Scope of the Development Project

The PGS software system is designed for educational settings, enabling students to create accounts for submitting their project forms. teachers can create accounts to grade the projects.

1.3 Definitions, acronyms, and abbreviations

Term	Definition
PGS	(Project Grading System) this project.
SDS	Software Design Specification
GUI	Graphical User Interface: A visual interface that allows users to interact with website.
Instructor	Person using the program to grade students' projects.
Student	Person using the program to submit project form.
Admin	Person managing the website

1.4 References

Dasgupta and D. W. Callahan. (2002). Development of improved tornado tracking device. IEEE 34th Southeastern Symposium on System Theory, pp. 363-365

1.5 Overview of Document

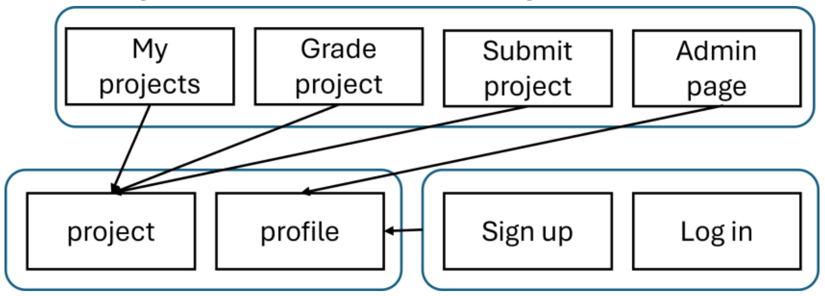
outlines the architecture, functionality, and technical specifications of a software system.

2. System Architecture Description

2.1 System Architecture

Development of the PGS will involve creating a website using Python. Each change will be logged and synchronized for team collaboration. The development environment will be set up on Windows.

2.1.1 High Level Architecture Diagram

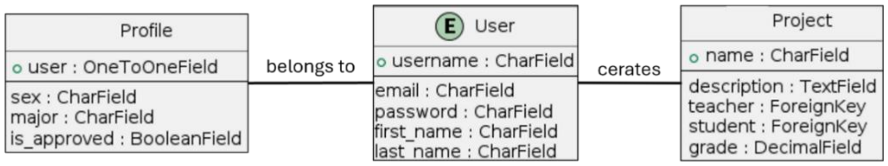


2.1.2 Architecture Narrative:

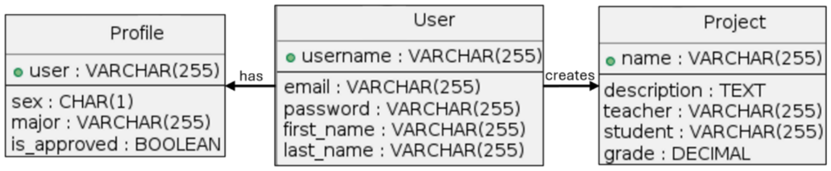
The architecture includes SQLite3 for lightweight database needs, Bootstrap for responsive web design, and Python for dynamic report generation. Communication relies on email for notifications, Django serves as middleware for web framework support, and implementation utilizes Python for backend logic with Django and JavaScript for frontend development. Using VSCode on Windows.

2.2 Database Components

2.2.1 Entity Relationship Diagram (ERD):



2.2.2 Database Table Definitions:



2.3 GUI Components with Layouts and Navigation

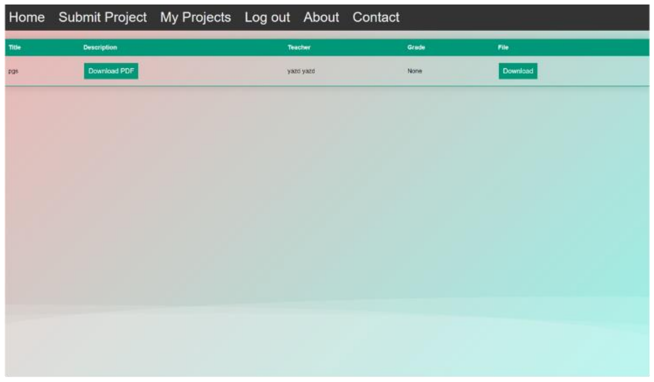
2.3.1 User Interface Issues

The PGS is a website designed with simplicity in mind with simple GUI interfaces.

2.3.2 List of User Interface Screens and Reports

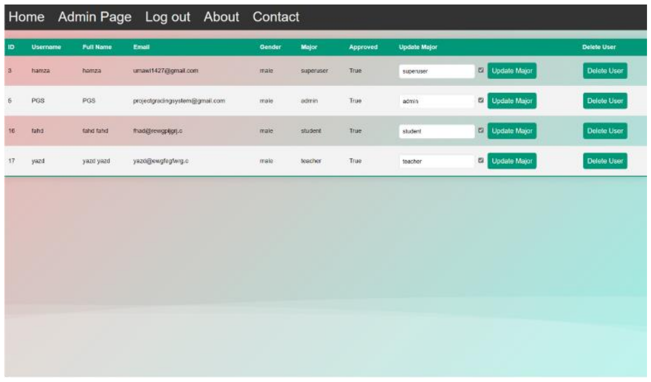
Contact, Login, sign up, Submit project, My projects, Grade projects, home pages.

2.3.3 User Interface Screens or Reports (Layouts)



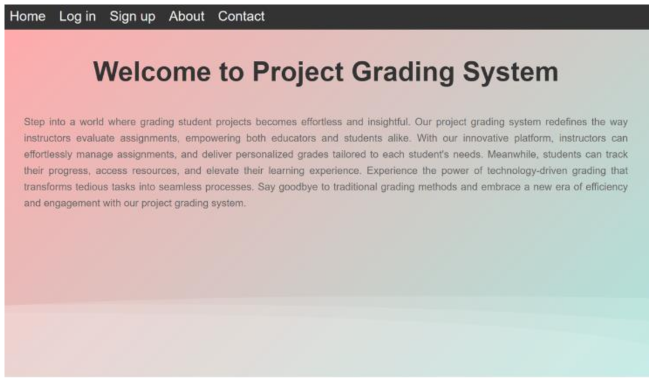
Title	Description	Teacher	Grade	File
pgs	Download PDF	yazd yazd	None	Download

My Projects Page: view grades of your projects, download your project file.



ID	Username	Full Name	Email	Gender	Major	Approved	Update Major	Delete User
3	hamza	hamza	umawi1427@gmail.com	male	superuser	True	superuser ✓ Update Major	Delete User
5	PGS	PGS	projectgradingsystem@gmail.com	male	admin	True	admin ✓ Update Major	Delete User
16	fahd	fahd fahd	fhad@rewgpljgrj.c	male	student	True	student ✓ Update Major	Delete User
17	yazd	yazd yazd	yazd@ewgfegfwrg.c	male	teacher	True	teacher ✓ Update Major	Delete User

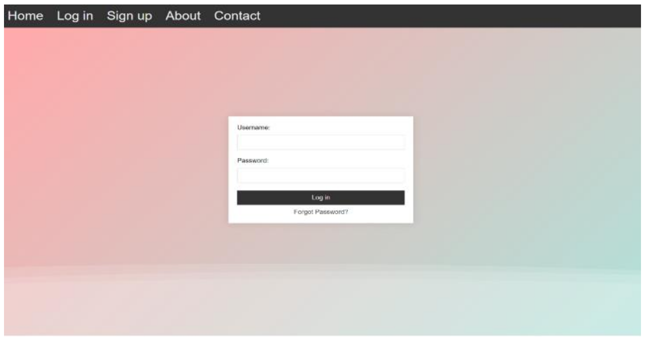
Admin Page: admin user approves and deletes accounts of users.



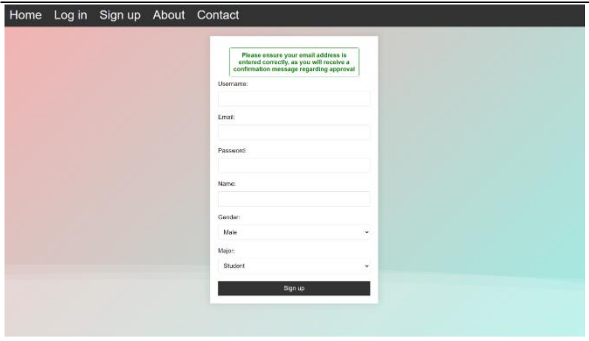
Welcome to Project Grading System

Step into a world where grading student projects becomes effortless and insightful. Our project grading system redefines the way instructors evaluate assignments, empowering both educators and students alike. With our innovative platform, instructors can effortlessly manage assignments, and deliver personalized grades tailored to each student's needs. Meanwhile, students can track their progress, access resources, and elevate their learning experience. Experience the power of technology-driven grading that transforms tedious tasks into seamless processes. Say goodbye to traditional grading methods and embrace a new era of efficiency and engagement with our project grading system.

Home Page: the main page of a website, featuring introductory descriptions.



Login Page: Enter your username and password, then click "Log in" to access your account. If you forgot your password, click "Forgot Password" to recover it.



Please ensure your email address is entered correctly, as you will receive a confirmation message regarding approval

Username:

Email:

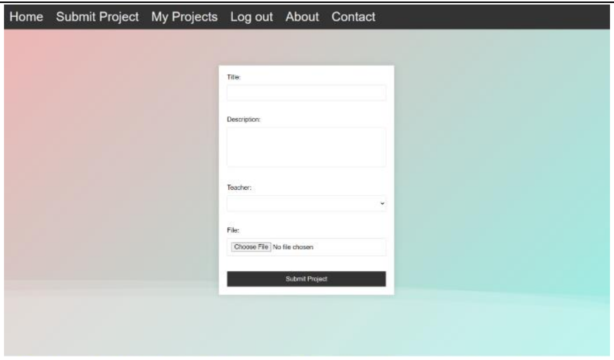
Password:

Name:

Gender: Male

Major: Student

Sign Up Page: Fill in your username, email, password, name, gender, and Select whether you are a student or instructor, then click "Sign up" to create a new account.



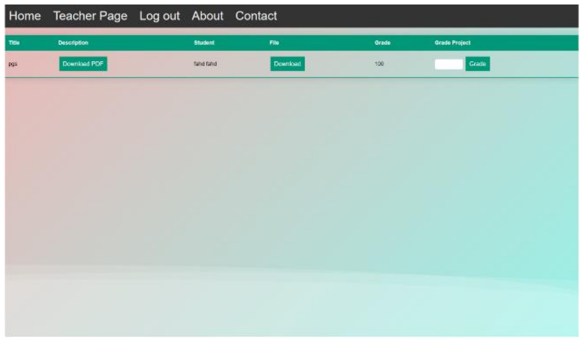
Title:

Description:

Teacher:

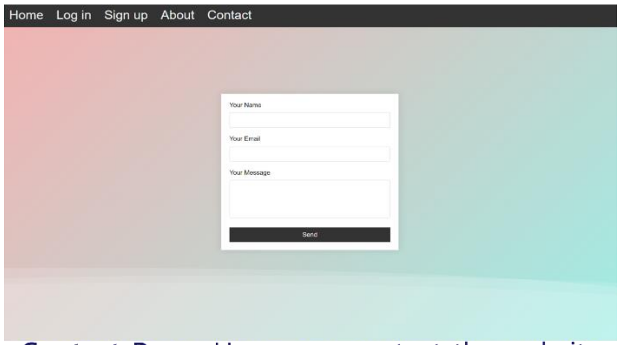
File: No file chosen

Submit Project Page: Enter the project name and description, select your teacher, upload your project file, and click "Submit Project".



Title	Description	Student	File	Grade	Grade Project
pgs	Download PDF	fahd fahd	Download	100	Grade

Grading project Page: Instructors use it to manage student submissions. They can download each submitted project from the "Download" column, review it, and then grade.



Your Name

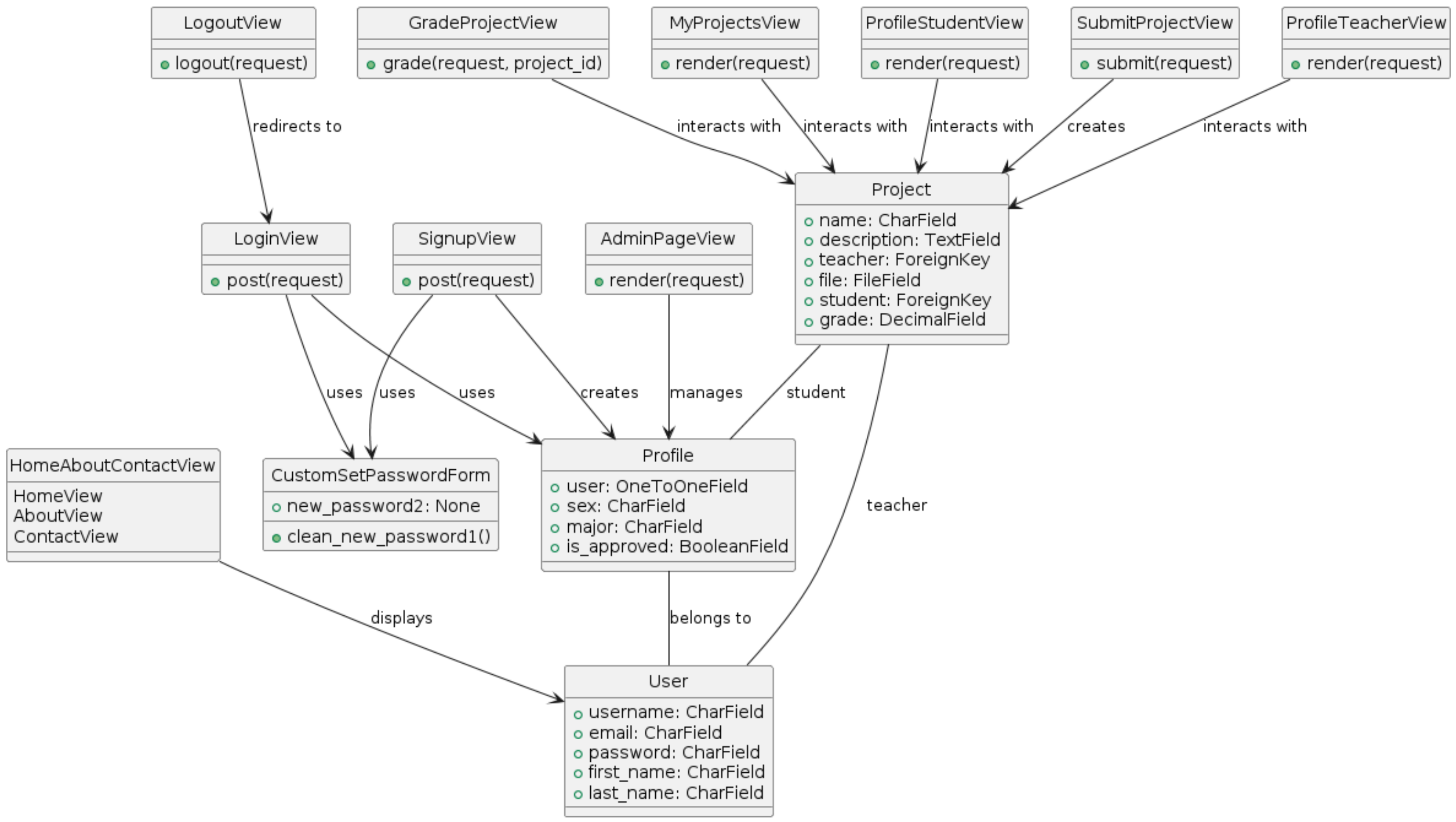
Your Email

Your Message

Contact Page: Users can contact the website administrators or support team by filling out their name, email, and message in the provided form and clicking the "Send" button.

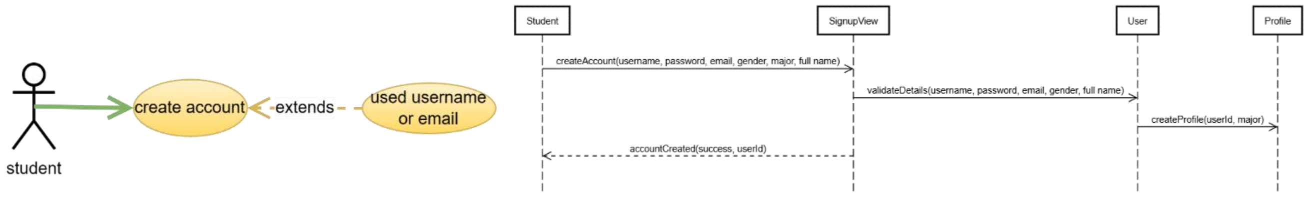
3. Detailed Description of System Components

3.1 Class Diagram

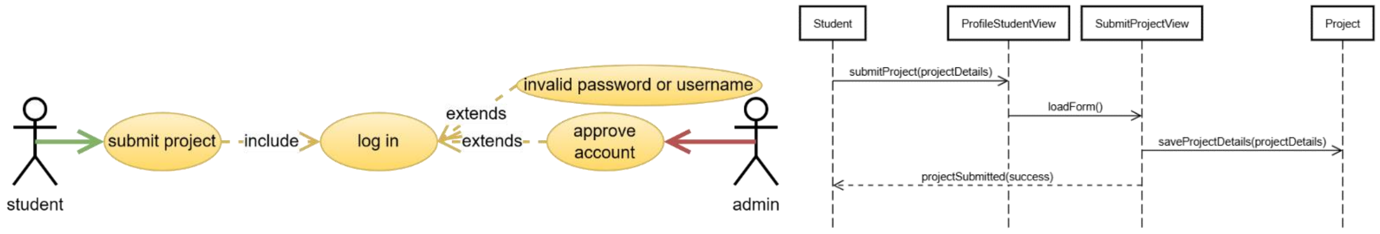


3.2 Use Cases with Sequence Diagrams

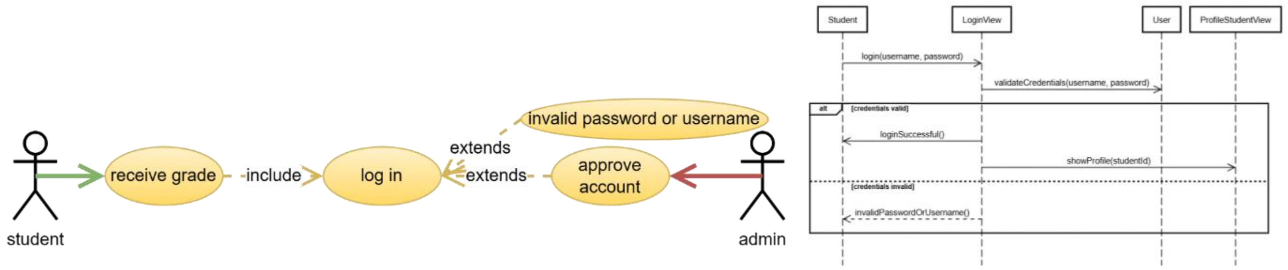
3.1.1 Sequence Diagram for Use Case 1


student
create account
extends
used username or email
Student
SignupView
User
Profile
createAccount(username, password, email, gender, major, full name)
validateDetails(username, password, email, gender, full name)
createProfile(userId, major)
accountCreated(success, userId)

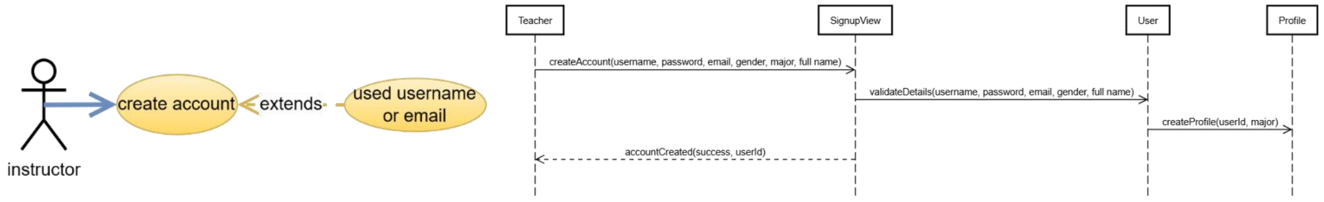
3.1.2 Sequence Diagram for Use Case 2



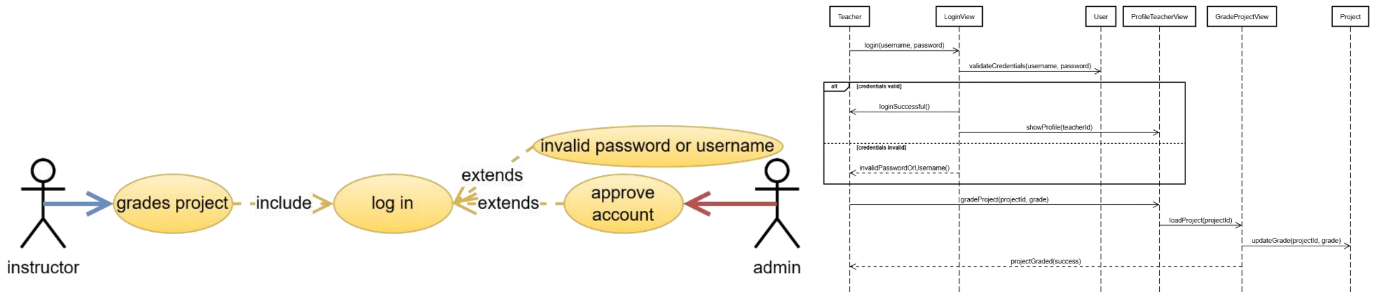
3.1.3 Sequence Diagram for Use Case



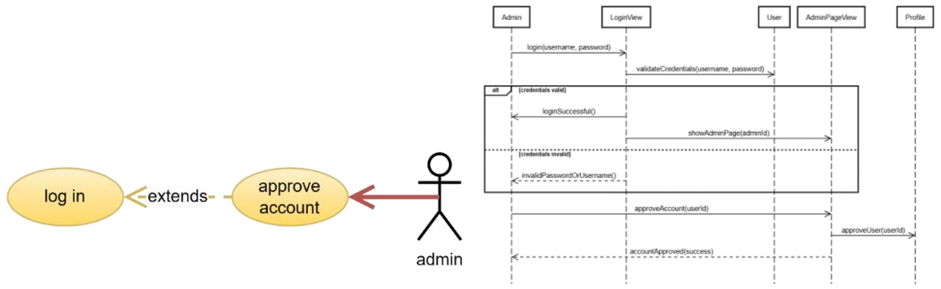
3.1.4 Sequence Diagram for Use Case 4


instructor
create account
extends
used username or email
Teacher
SignupView
User
Profile
createAccount(username, password, email, gender, major, full name)
validateDetails(username, password, email, gender, full name)
createProfile(userId, major)
accountCreated(success, userId)

3.1.5 Sequence Diagram for Use Case 5



3.1.6 Sequence Diagram for Use Case 6



4. Design Decisions and Tradeoffs

4.1 Web Platform

We chose to develop the PGS as a website using Python (Django Framework).

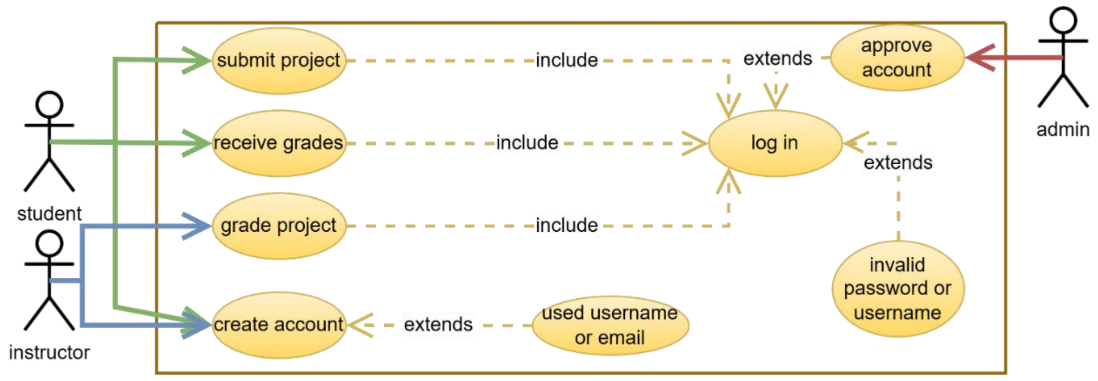
4.2 Feature Selection

Initially, we considered various features for PGS. We decided to focus on core functionalities: allowing students to submit homework and projects, and enabling instructors to grade.

4.3 Modular Design

We adopted a modular design for PGS, making each feature a separate module. For easier updates.

Appendix A: Use-Case Diagram



Appendix B: Data Dictionary

B1: Use-Case Description

1- Student Use-Cases: The student has the following sets of use cases:

1. **Create account:**

Brief Description / Goal: The "Create Account" functionality enables students to register for a new account on the PGS website. It allows students to submit their projects after approvement.

Pre-conditions:

1. The student must have computer connected to the internet with browser.

Initial Step-By-Step Description / scenario:

1. **Accessing Registration Page:** The student navigates to the "sign up" page.
2. **Providing User Information:** The student fills out the registration form, providing necessary details such as: Full Name, Email Address, Password.
3. **Submitting Registration Form:** the student clicks on the "sign up" button.
4. **Confirmation:** Upon successful registration, the system confirms the creation of the account.
5. **Awaiting approval:** The student is awaiting administrative approval for their chosen major to access their account. They will receive an email notification regarding approval from the PGS team.

Post-conditions:

1. The student's account information is securely stored in the PGS database, and they receive confirmation of successful registration.
2. The student can log in using their provided credentials after the admin approves their majors.

Exceptions:

1. **Used username or email:** If the username or email address is already used, the student is prompted to provide a different username or email address.
2. **Submit project Use-Case:**

Brief Description / Goal: submit project allows students to upload projects for grading by instructors.

Pre-conditions:

1. The student must have a valid and approved account and be logged in to the PGS website.
2. The student must have the file of project to upload.

Initial Step-By-Step Description / scenario:

1. **Accessing the "Submit Project" Section:** The student navigates to the "Submit Project" section within the PGS website after logging in.
2. **Entering project Details:** The student enters project required details.
3. **Uploading project file:** The student uploads project file.
4. **Submission Section:** student clicks on the "Submit project" button.

Post-conditions:

1. The project details and uploaded files are successfully stored in the PGS database, linked to the student's account for future reference and grading.
2. The student receives confirmation of successful project submission

Exceptions:

1. **Invalid password or username:** If the username or password provided during the login process is incorrect, an error message is displayed, prompting the student to enter the correct credentials.
2. **Account not approved:** If the administrator has not yet approved the user's major, the user will be unable to log into the system.

2. **recieve grades Use-Case:**

Brief Description / Goal: this functionality enables students to view grades provided by instructors.

Pre-conditions:

1. The student must have submitted a project using the "Submit Project" functionality.
2. The instructor must have graded the submitted project.

Initial Step-By-Step Description:

1. **Accessing the "my projects" Section:** The student navigates to the "My Projects" section.
2. **Reviewing Grades:** The student read through the grades provided by the instructor.

Post-conditions:

1. The student successfully accesses and reviews their grades provided by the instructor.

Exceptions:

1. **Invalid password or username:** If the username or password provided during the login process is incorrect, an error message is displayed, prompting the student to enter the correct credentials.
2. **Account not approved:** If the administrator has not yet approved the user's major, the user will be unable to log into the system.

2- **Instructor use-cases:** The teacher has the following sets of use cases:

1. **Create account:**

Brief Description / Goal: The "Create Account" functionality enables instructors to register for a new account on the PGS website. It allows instructors to view and grade students' projects.

Pre-conditions:

1. The instructor must have computer connected to the internet with browser.

Initial Step-By-Step Description / scenario:

1. **Accessing Registration Page:** The instructor navigates to the "sign up" page.
2. **Providing User Information:** The instructor fills out the registration form, providing necessary details such as: Full Name, Email Address, Password.
3. **Submitting Registration Form:** The instructor clicks on the "sign up" button.
4. **Confirmation:** Upon successful registration, the system confirms the creation of the account.
5. **Awaiting approval:** The instructor is awaiting administrative approval for their chosen major to access their account. They will receive an email notification regarding approval from the PGS team.

Post-conditions:

1. The instructor's account information is securely stored in the PGS database, and they receive confirmation of successful registration.
2. The instructor can log in using their provided credentials after the admin approves their majors.

Exceptions:

1. **Used username or email:** If the username or email address is already used, the instructor is prompted to provide a different username or email address.

2. **Grade project Use-Case:**

Brief Description / Goal: "Grade Project" functionality enables instructors to grade projects for students.

Pre-conditions:

1. The instructor must have a valid and approved account and be logged in to the PGS.
2. Projects submitted by students must exist in the system.
3. The "Teacher Page" section must be accessible and available within the PGS website.
4. The instructor must have selected a project to grade.

Initial Step-By-Step Description:

1. **Accessing Submission Section:** The instructor navigates to the "Teacher Page".
2. **Selecting Project to Grade:** The instructor chooses the specific project to assign grades.
3. **Entering Grades:** The instructor inputs grades for the selected project submission. Which enable the student to view it.

Post-conditions:

1. The instructor successfully assigns grades for the selected project submission.
2. Students receive the grades of their projects.

Exceptions:

1. **Invalid password or username:** If the username or password provided during the login process is incorrect, an error message is displayed, prompting the student to enter the correct credentials.

2. **Account not approved:** If the administrator has not yet approved the user's major, the user will be unable to log into the system.

3- **Admin use-cases:** The Admin has the following use case:

1. **Approval and Deletion of User Accounts Based on Major use-case:**

Brief Description / Goal: approving or rejecting selected major of user and managing account deletions.

Pre-conditions:

1. Users have submitted their chosen major for approval.
2. Admin privileges are accessible to manage user accounts.

Initial Step-By-Step Description:

1. Admin selects the pending major approval section. Then approve or reject the request.
2. The admin chooses the account they wish to remove and proceeds with its deletion.

Post-conditions:

1. After approval, the system will send a notification to the user's email confirming the acceptance of their chosen major, granting them access to their accounts.
2. Deleted accounts are permanently removed from the system.

B2: Actor Descriptions

Student: A user who registers on the PGS to submit projects for grading and view their grades.

instructor: A user who registers on the PGS to grade projects submitted by students.

Admin: A user is responsible for approving student and instructor accounts. Managing system.

B3: Class Descriptions

User: represents a user with these attributes: username, email, password, full name, all are [CharFields].

Profile: Contains additional information about a user. with these attributes: user, [OneToOneField] to the User class; sex, major [CharFields]; and is_approved, which is a BooleanField. This class belongs to a User.

Project: Represents a project created by students. with these attributes: name, [CharField]; description [TextField]; teacher [ForeignKey]; file [FileField]; student [ForeignKey]; grade [DecimalField]

CustomSetPasswordForm: A form for setting a new password. has attribute [new_password2], which is of type None. It also has the method clean_new_password1().

HomeAboutContactView: home, about, and contact pages classes.

LogoutView: logging out. has a method logout(request), redirects to LoginView.

LoginView: logging in. It has the method post(request).

SignupView: signing up. It has the method post(request). Belongs to user and profile

AdminPageView: the admin page. It has the method render(request).

MyProjectsView: displaying a user's projects. It has the method render(request).

ProfileStudentView: displaying a student's profile. It has the method render(request).

SubmitProjectView: submitting a project. It has the method submit(request).

ProfileTeacherView: displaying a teacher's profile. It has the method render(request).

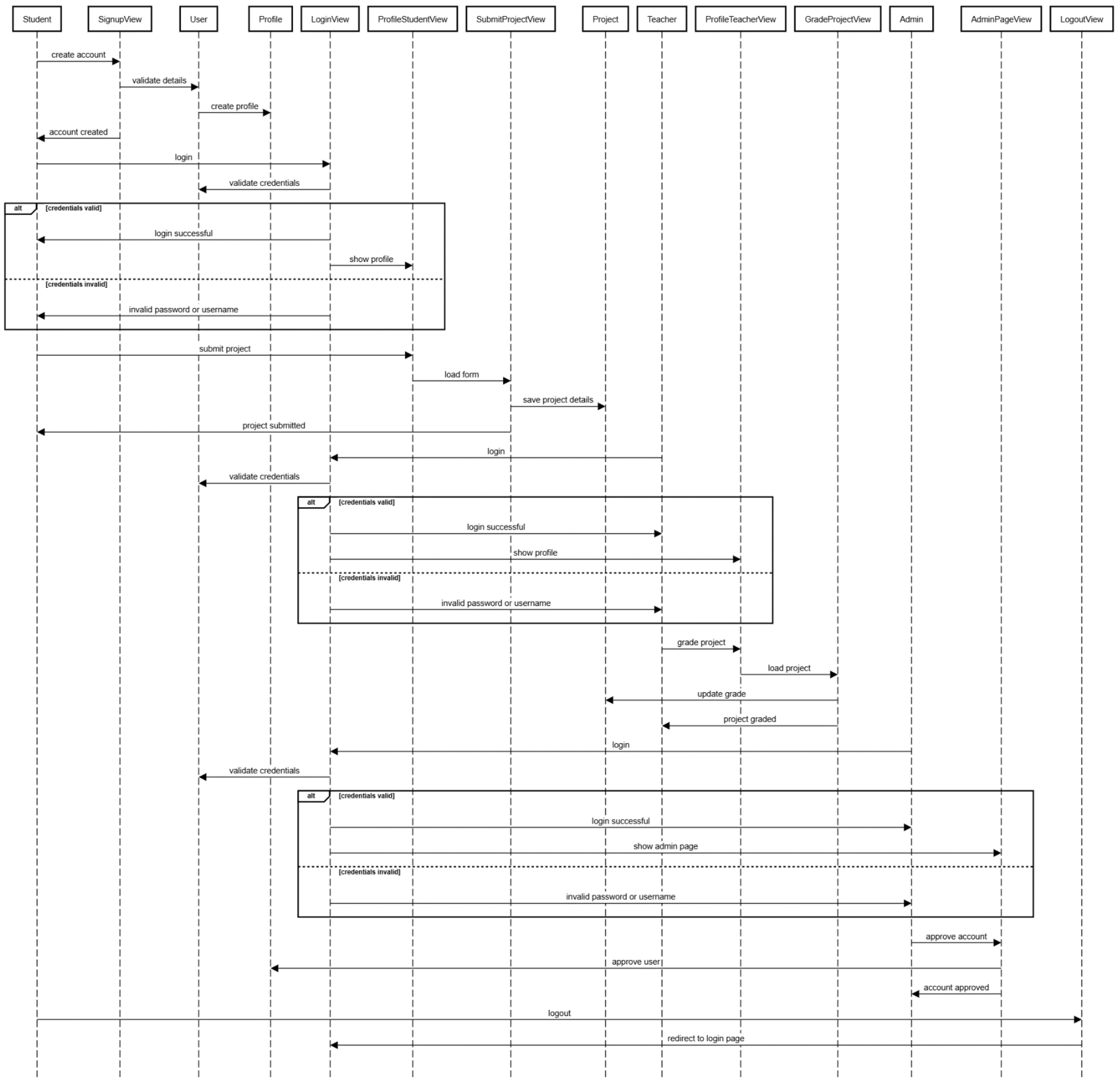
GradeProjectView: grading a project. It has the method grade(request, project_id).

Appendix C: List of Inputs and Outputs

Inputs: Students and instructors must provide registration data, including their names, email addresses, and passwords. For project submissions, students need to upload their project files along with the title, description and teacher name. Instructors input grades for each project. Additionally, login credentials and contact form details, such as name, email, and message, are also required inputs.

Outputs: Students receive a dashboard displaying their submitted projects ang grades. Instructors access a dashboard listing all student submissions with options to download project. Confirmation messages are provided for successful actions like contact form submissions, error messages alert users to issues such as invalid credentials or failed submissions. Email notifications inform users of registration.

Appendix D: Sequence Diagram


Student
SignupView
User
Profile
LoginView
ProfileStudentView
SubmitProjectView
Project
Teacher
ProfileTeacherView
GradeProjectView
Admin
AdminPageView
LogoutView
create account
validate details
create profile
account created
login
validate credentials
alt
[credentials valid]
login successful
show profile
[credentials invalid]
invalid password or username
submit project
load form
save project details
project submitted
login
validate credentials
alt
[credentials valid]
login successful
show profile
[credentials invalid]
invalid password or username
grade project
load project
update grade
project graded
login
validate credentials
alt
[credentials valid]
login successful
show admin page
[credentials invalid]
invalid password or username
approve account
approve user
account approved
logout
redirect to login page