Advising Appointment Scheduling System

View Team:	Sarah, Gungun, Aidan
Model Team:	Cooper, Ethan, Avinash, Anshika, Wyatt, Agampaul
SQA Team:	Joshua, Austin, Christelle
CM Team:	James, Landan, Stephen

Brief AASS Overview

The Advising Appointment Scheduling System is a web-based application designed for use by students and faculty for creating and managing advising appointments.

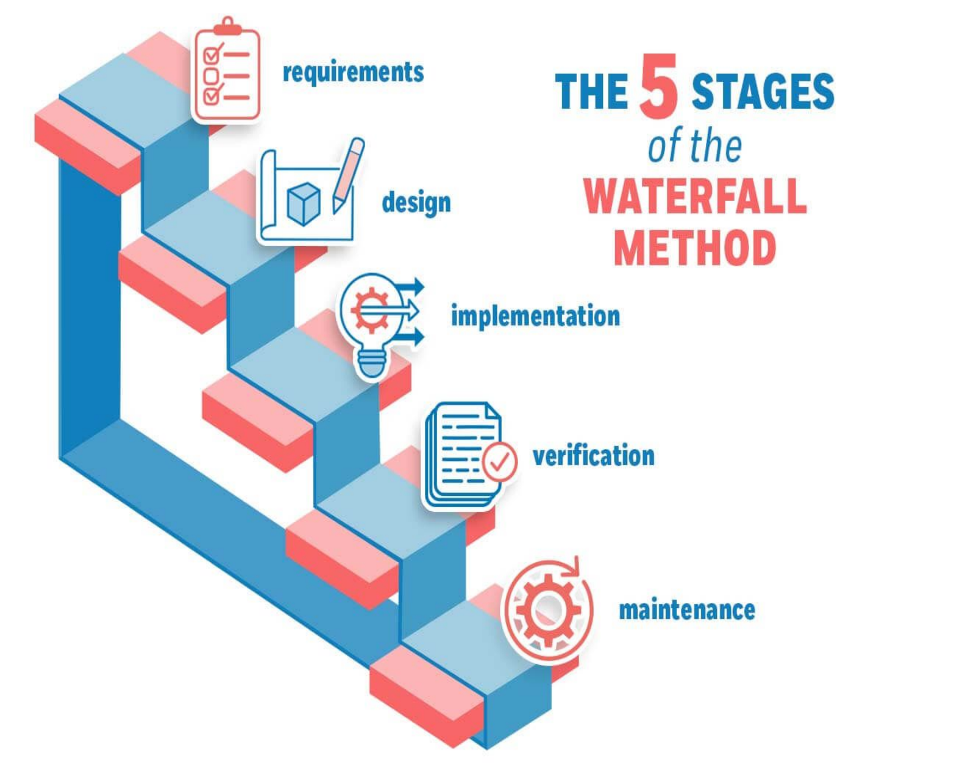
Key Elements:

- Faculty
- Students
- Admin
- Appointments

Team Organization

Process Model Used: Waterfall Process Model

1. Requirements
2. Designing
3. Implementation
4. Testing
5. Maintenance (After deployment)



Team Organization

Tools used:

- Communication
  - Discord
- Development
  - C++
  - Wt
  - SQLite 3
  - OpenBSD
  - HTML, XHTML, CSS, JavaScript

Team Organization

Training:

- Wt Documentation
- Bootstrap 5 Documentation
- XHTML Documentation
- SQLite 3 Documentation
- Software Engineering: A Practitioner’s Approach

Configuration Management Plan for AASS

Purpose: To ensure consistent handling and documentation of all changes to software and associated products.

Scope: Applicable to all team members involved in the AASS project.

Key Definitions:

- **Configuration Item (CI):** Any aggregated work product managed as a single entity within the configuration management process.
- **Baseline:** The formally reviewed and agreed-upon version of a CI that serves as the base for future developments.
- **Gitea:** Software used for version control, issue tracking, and other collaborative features.

Configuration Management Overview

Team Structure

- **Sub-Teams:** Front-End, Back-End, Quality Assurance, Change Management
- **Roles & Responsibilities:** Change Management team enforces CMP, ensures compliance.

Configuration Items

- **Examples:** Software Requirements Specification, Test Plans, Design Documents.
- **Management Process:** Storage, access, and revision handled via the Software Development Library (SDL).

Control Processes and Reporting

Change Control Tools

- **Primary Tool:** Gitea
- **Features:** Version control, issue tracking, merge permissions, safeguards and approval requirements.

Change Request Procedure

- **Submission:** By any team member via Gitea.
- **Review:** By Change Management team or other team members.
- **Approval:** Two approvals are required along with merge action from any Product Owner.

Reporting

- **Status Reports:** Compiled by Product Owners, detail progress and issues.
- **Issue Tracking:** Managed through Gitea, includes ticket creation and resolution tracking.

Milestones, Training, and Conclusion

Milestones

- **Tracking:** Milestones were tracked by each team individually. The Kanban boards in Gitea represent these milestones visually.
- **Significance:** Important for tracking progress and meeting stakeholder expectations.
- **Examples:** Completion of issues, development progress, etc.

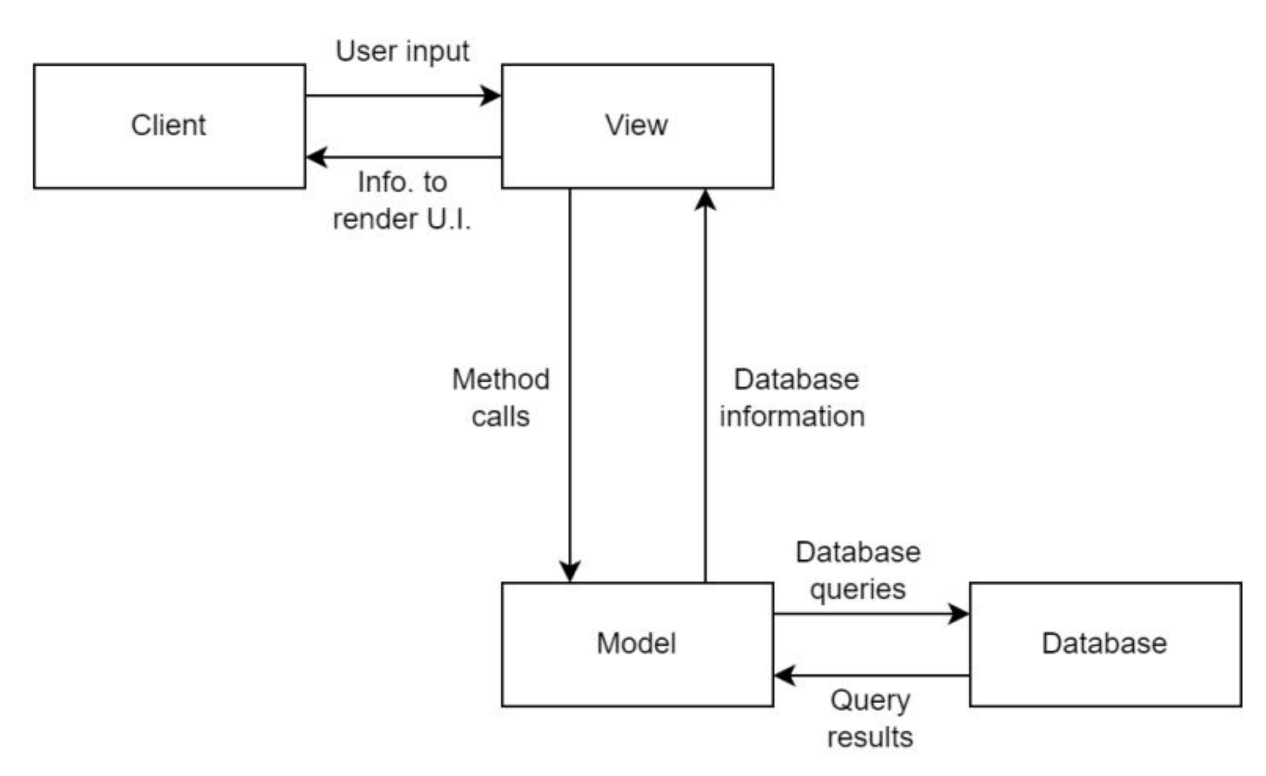
Training

- **Topics:** Specific technologies and methodologies relevant to each team's focus.
- **Purpose:** Ensures all team members are proficient and can effectively contribute to their tasks.

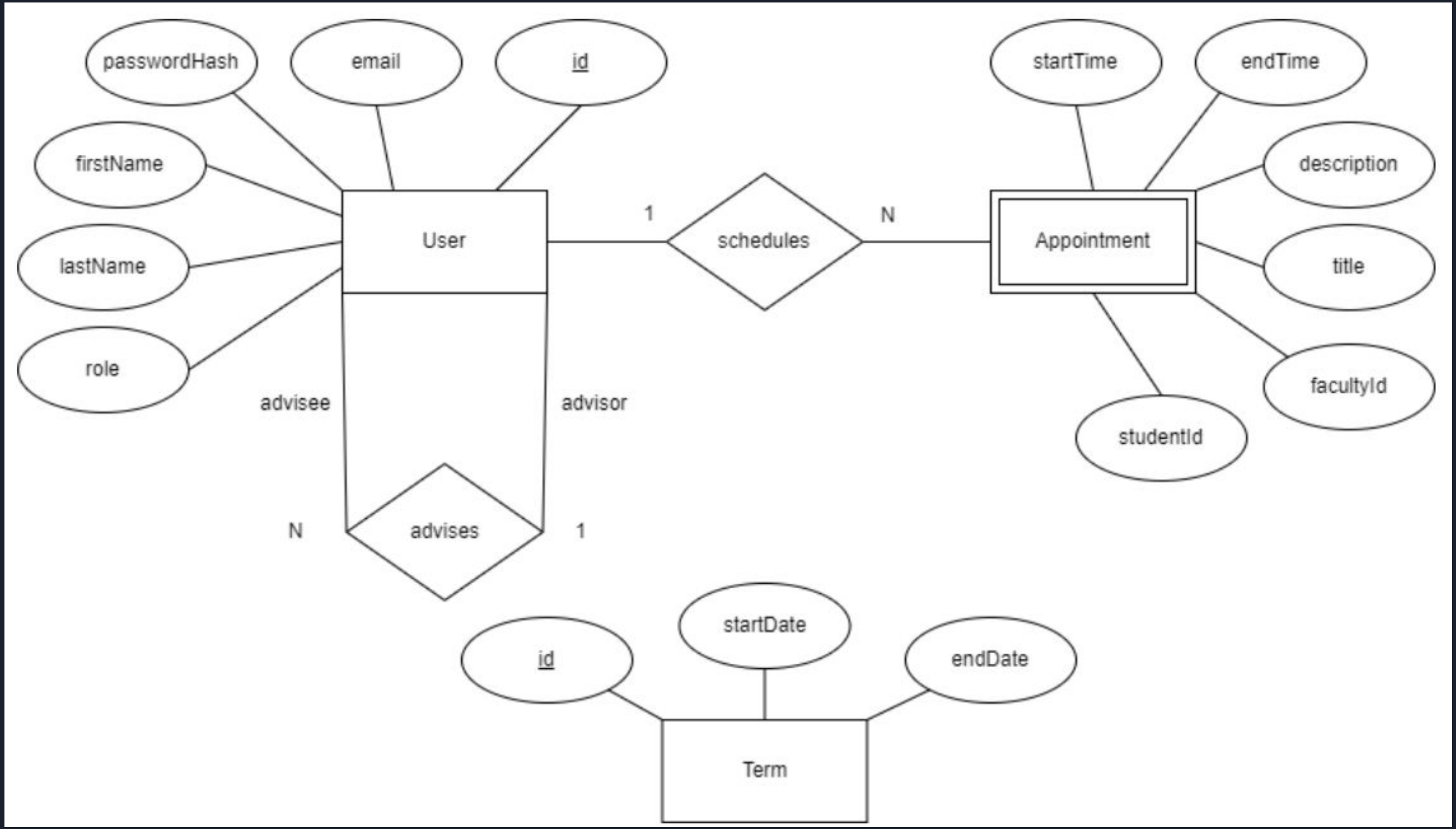
Software Quality Assurance Plan

- The SQA plan was created to ensure that
  - The software conforms to the requirements
  - The software quality is kept
  - Reduce the number of bugs from testing
- Applies to the members of the AASS team
- Testing done includes:
  - Unit tests
  - Regression tests
  - Integration tests

Software Design



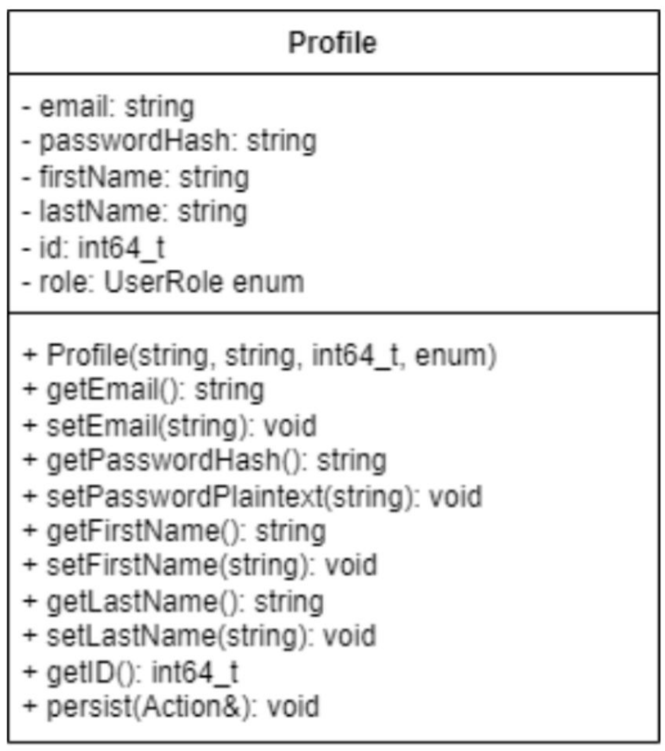
Design Artifacts
(Model Entity Relationship Diagram)



Design Artifacts
(Profile Model)

- **Purpose of this Model:**
  - This model serves as a representation of a “User” mentioned in the “Entity Relationship” diagram.

- Data Items:
  - Role is an enumerated type containing Student, Admin and Faculty.
  - FirstName, LastName.
  - The Email, Password and ID are a part of the Witty Auth Table.



Profile

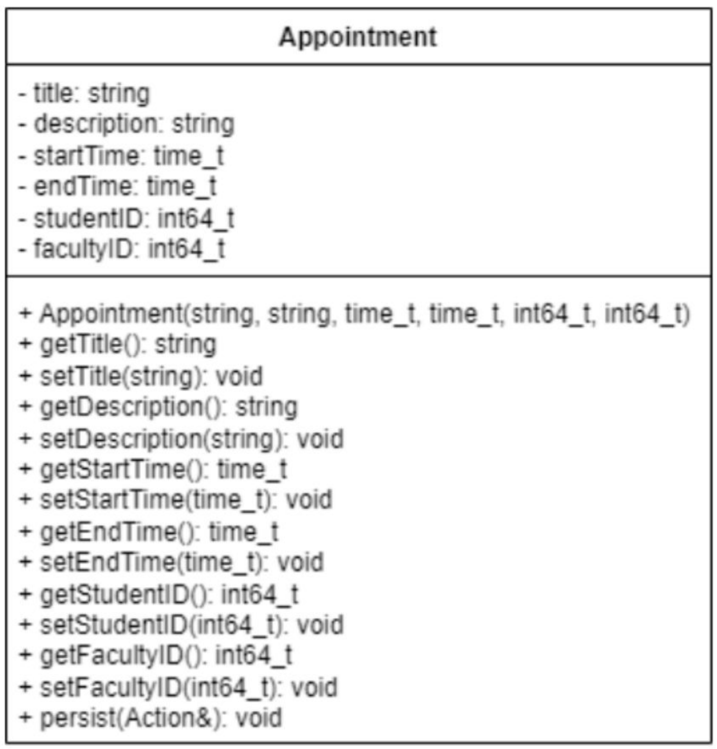
- email: string
- passwordHash: string
- firstName: string
- lastName: string
- id: int64_t
- role: UserRole enum

+ Profile(string, string, int64_t, enum)
+ getEmail(): string
+ setEmail(string): void
+ getPasswordHash(): string
+ setPasswordPlaintext(string): void
+ getFirstName(): string
+ setFirstName(string): void
+ getLastName(): string
+ setLastName(string): void
+ getID(): int64_t
+ persist(Action&): void

Design Artifacts
(Appointment Model)

- **Purpose of this Model:**
  - This model serves as a representation of all data relevant to a single scheduled appointment.

- **Data Items:**
  - title: length [1 to 30]]
  - description: length [1 to 100]
  - startTime and endTime: Stores UNIX timestamp.
  - studentID and facultyID: profileID

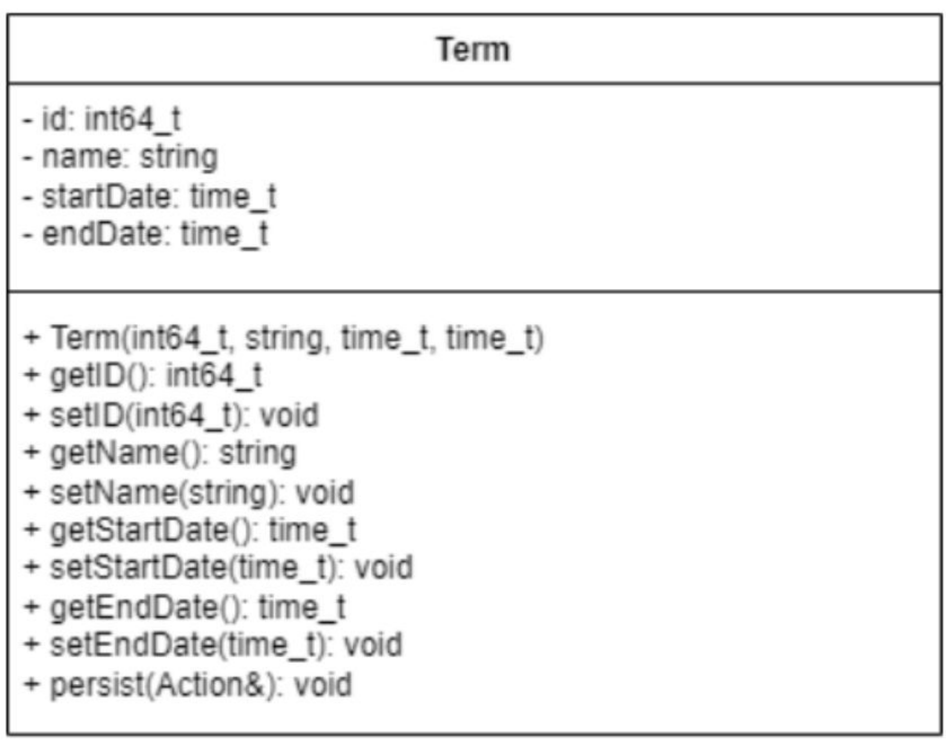


```
Appointment

- title: string
- description: string
- startTime: time_t
- endTime: time_t
- studentID: int64_t
- facultyID: int64_t

+ Appointment(string, string, time_t, time_t, int64_t, int64_t)
+ getTitle(): string
+ setTitle(string): void
+ getDescription(): string
+ setDescription(string): void
+ getStartTime(): time_t
+ setStartTime(time_t): void
+ getEndTime(): time_t
+ setEndTime(time_t): void
+ getStudentID(): int64_t
+ setStudentID(int64_t): void
+ getFacultyID(): int64_t
+ setFacultyID(int64_t): void
+ persist(Action&): void
```

Design Artifacts
(Term Model)

- **Purpose of this Model:**
  - This model stores data related to academic terms in which appointments can be scheduled.

- Data Types:
  - startTime and endTime: store UNIX timestamp representing the start of the term and end of the term respectively.
  - name : stores a short string with a length in the range [1,30].
  - id: stores a positive integer in the range [0,(2^63 – 1)].



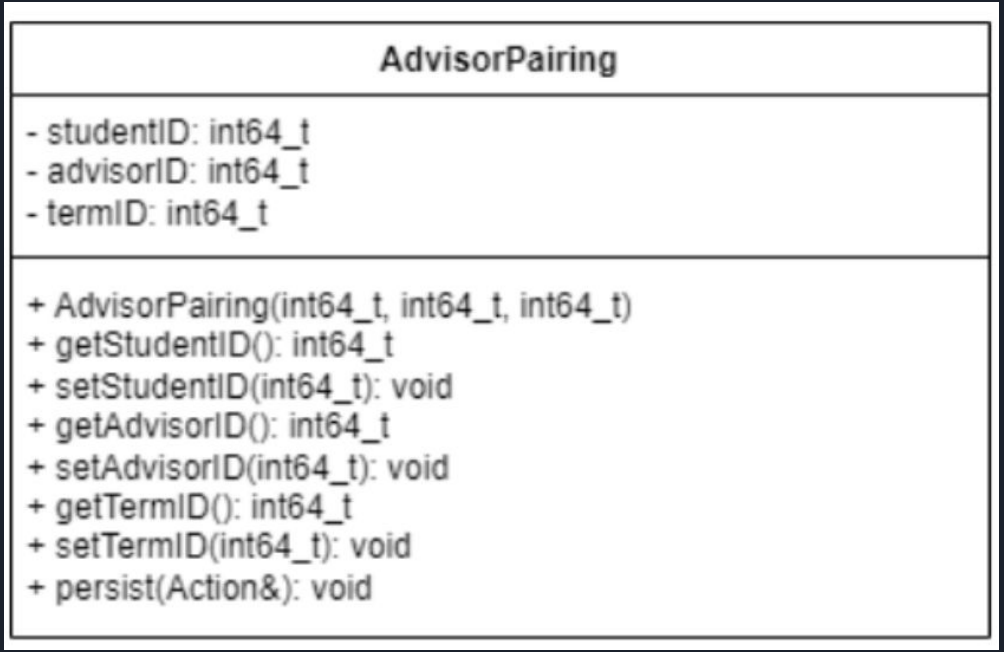
Term

- id: int64_t
- name: string
- startDate: time_t
- endDate: time_t

+ Term(int64_t, string, time_t, time_t)
+ getID(): int64_t
+ setID(int64_t): void
+ getName(): string
+ setName(string): void
+ getStartDate(): time_t
+ setStartDate(time_t): void
+ getEndDate(): time_t
+ setEndDate(time_t): void
+ persist(Action&): void

Design Artifacts
(AdvisorPairing Model)

- **Purpose of this Model:**
  - This model stores IDs that indicate which faculty member is assigned to each student inside of a term.
- **Data Items:**
  - Student ID: This stores the ID of a student in the pairing.
  - Advisor ID: This stores the ID of a faculty member in the pairing.
  - Term ID: This stores the ID of the term in which the pairing is valid.



AdvisorPairing

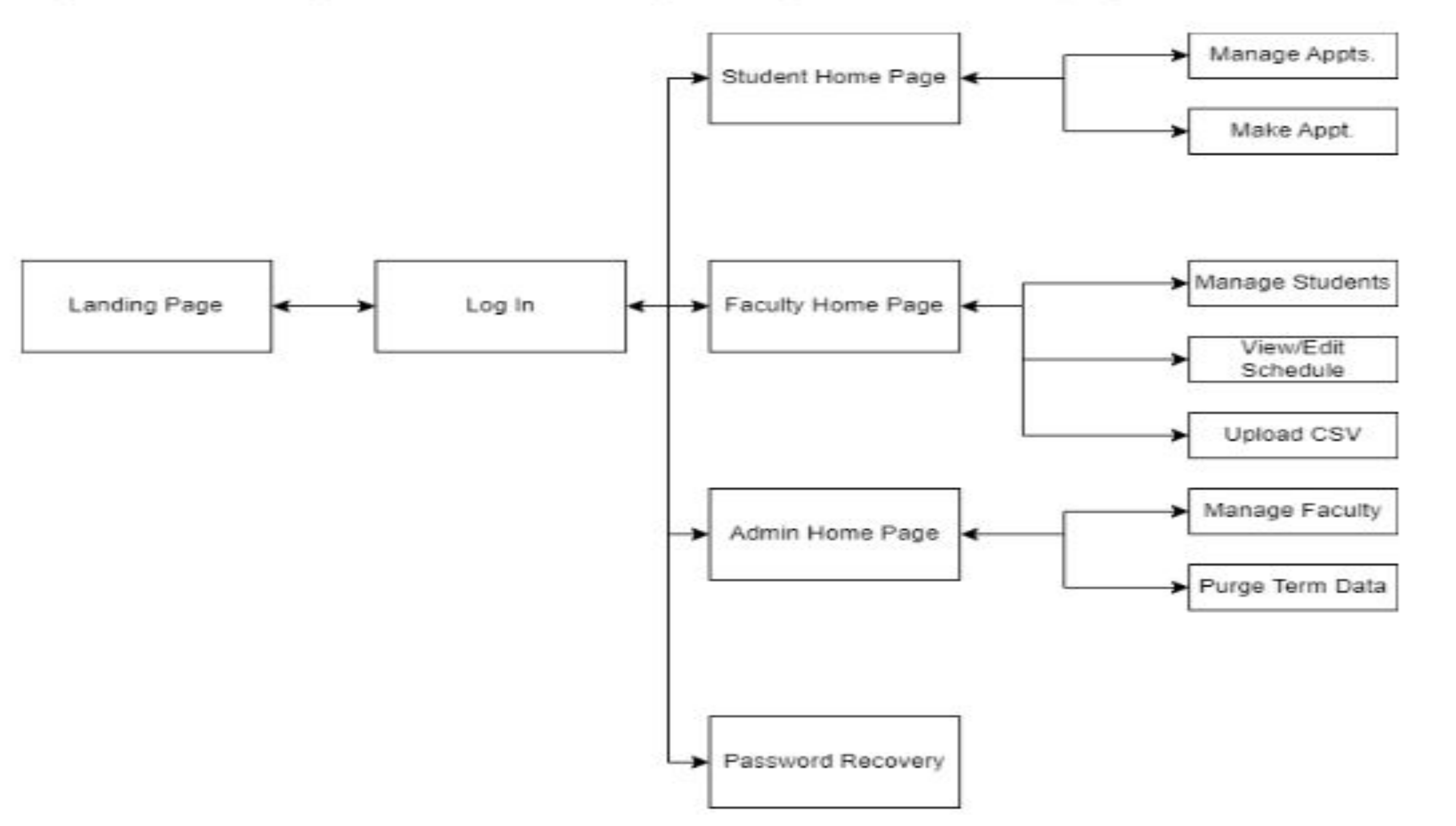
```
- studentID: int64_t
- advisorID: int64_t
- termID: int64_t

+ AdvisorPairing(int64_t, int64_t, int64_t)
+ getStudentID(): int64_t
+ setStudentID(int64_t): void
+ getAdvisorID(): int64_t
+ setAdvisorID(int64_t): void
+ getTermID(): int64_t
+ setTermID(int64_t): void
+ persist(Action&): void
```

Interface & Wrapper Classes

- Each model has a class that abstracts queries for the respected objects.
    - Constructor takes in Wt::Dbo::Session object to handle database operations.

- Interface designed so other objects and views can be tested using a dummy class.

Software Architectural Design



Software Architectural Design (Student)

- Student Homepage
  - + Presents the student user with the navigation to the manage and make appointment pages.
- Manage Appointments
  - + Provides the ability for the student user to interact with and manage their booked appointments.
- Make Appointments
  - + A screen for the student user in which they can book appointments w/their advisor.
  - + Includes ability to book a day & custom time range for appts.

Software Architectural Design (Faculty)

MANAGE STUDENTS

- Create new student profiles.
- Edit existing student profiles.
- Delete existing student profiles.

VIEW/EDIT SCHEDULE

- View existing appointments.
- Edit own appointments.
- Set unavailable times.

UPLOAD CSV

- Upload CSV for student profiles.
- Upload CSV for scheduling data.
- Display error message for incorrect CSV.

Software Architectural Design (Admin)

Admin Home Page

Has the option to navigate to the “Manage Faculty” page or “Purge Old Terms” page

Manage Faculty Page

Is able to create, edit, and delete a faculty member

Purge Old Terms Page

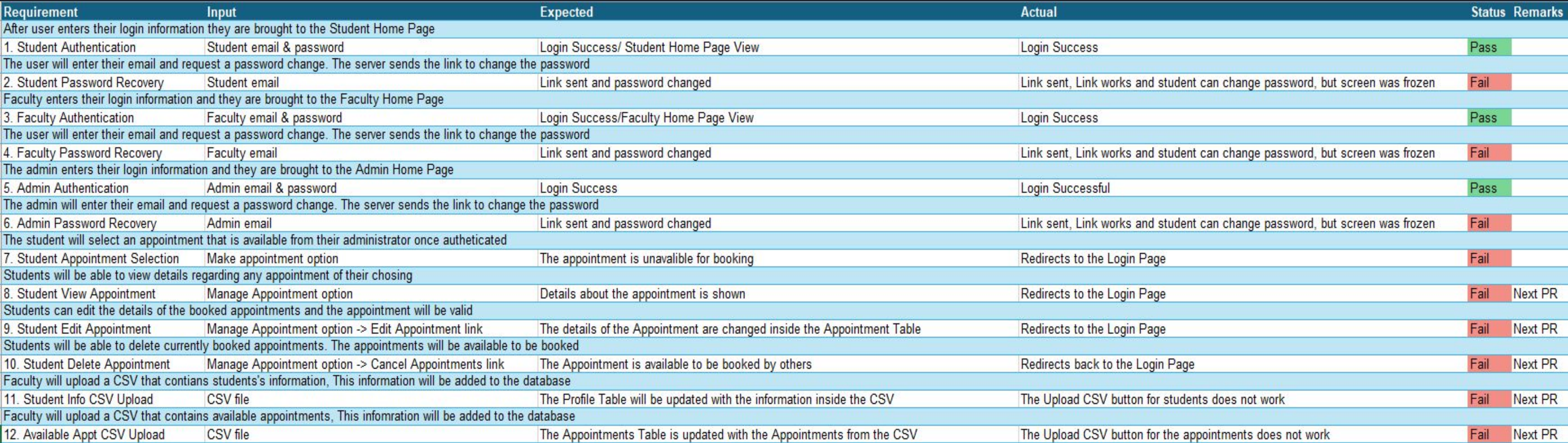
Gives option to delete a term in the database that is past completion

Testing Approach

- Unit tests were conducted by the developers of the back-end and front-end team
- The first round of testing by the SQA team was smoke testing.
  - Not every component was finalized
- The second and final round of testing by the SQA team was integration testing
  - Every Requirement was tested
- SQA tests include:
  - Requirement
  - Input
  - Expect Output
  - Actual Output
  - Status
  - Remarks

Smoke Testing

- Superficial Smoke testing took place consistently over the latter half of development
- In depth smoke testing has taken place during the final stages of development as we prepared for integration testing



Requirement	Input	Expected	Actual	Status	Remarks
After user enters their login information they are brought to the Student Home Page					
1. Student Authentication	Student email & password	Login Success/ Student Home Page View	Login Success	Pass	
The user will enter their email and request a password change. The server sends the link to change the password					
2. Student Password Recovery	Student email	Link sent and password changed	Link sent, Link works and student can change password, but screen was frozen	Fail	
Faculty enters their login information and they are brought to the Faculty Home Page					
3. Faculty Authentication	Faculty email & password	Login Success/Faculty Home Page View	Login Success	Pass	
The user will enter their email and request a password change. The server sends the link to change the password					
4. Faculty Password Recovery	Faculty email	Link sent and password changed	Link sent, Link works and student can change password, but screen was frozen	Fail	
The admin enters their login information and they are brought to the Admin Home Page					
5. Admin Authentication	Admin email & password	Login Success	Login Successful	Pass	
The admin will enter their email and request a password change. The server sends the link to change the password					
6. Admin Password Recovery	Admin email	Link sent and password changed	Link sent, Link works and student can change password, but screen was frozen	Fail	
The student will select an appointment that is available from their administrator once autheticated					
7. Student Appointment Selection	Make appointment option	The appointment is unavalible for booking	Redirects to the Login Page	Fail	
Students will be able to view details regarding any appointment of their chosing					
8. Student View Appointment	Manage Appointment option	Details about the appointment is shown	Redirects to the Login Page	Fail	Next PR
Students can edit the details of the booked appointments and the appointment will be valid					
9. Student Edit Appointment	Manage Appointment option -> Edit Appointment link	The details of the Appointment are changed inside the Appointment Table	Redirects to the Login Page	Fail	Next PR
Students will be able to delete currently booked appointments. The appointments will be available to be booked					
10. Student Delete Appointment	Manage Appointment option -> Cancel Appointments link	The Appointment is available to be booked by others	Redirects back to the Login Page	Fail	Next PR
Faculty will upload a CSV that contians students's information, This information will be added to the database					
11. Student Info CSV Upload	CSV file	The Profile Table will be updated with the information inside the CSV	The Upload CSV button for students does not work	Fail	Next PR
Faculty will upload a CSV that contains available appointments, This infomration will be added to the database					
12. Available Appt CSV Upload	CSV file	The Appointments Table is updated with the Appointments from the CSV	The Upload CSV button for the appointments does not work	Fail	Next PR

Integration Test

Activities: Tested every functionality of the system

Who: Joshua and Austin

How: Ran the AASS through CS server for testing

When: 5/6/2024 - 5/7/2024

Results:

Available Appt CSV Upload - Failed

Student Notification System - Not implemented

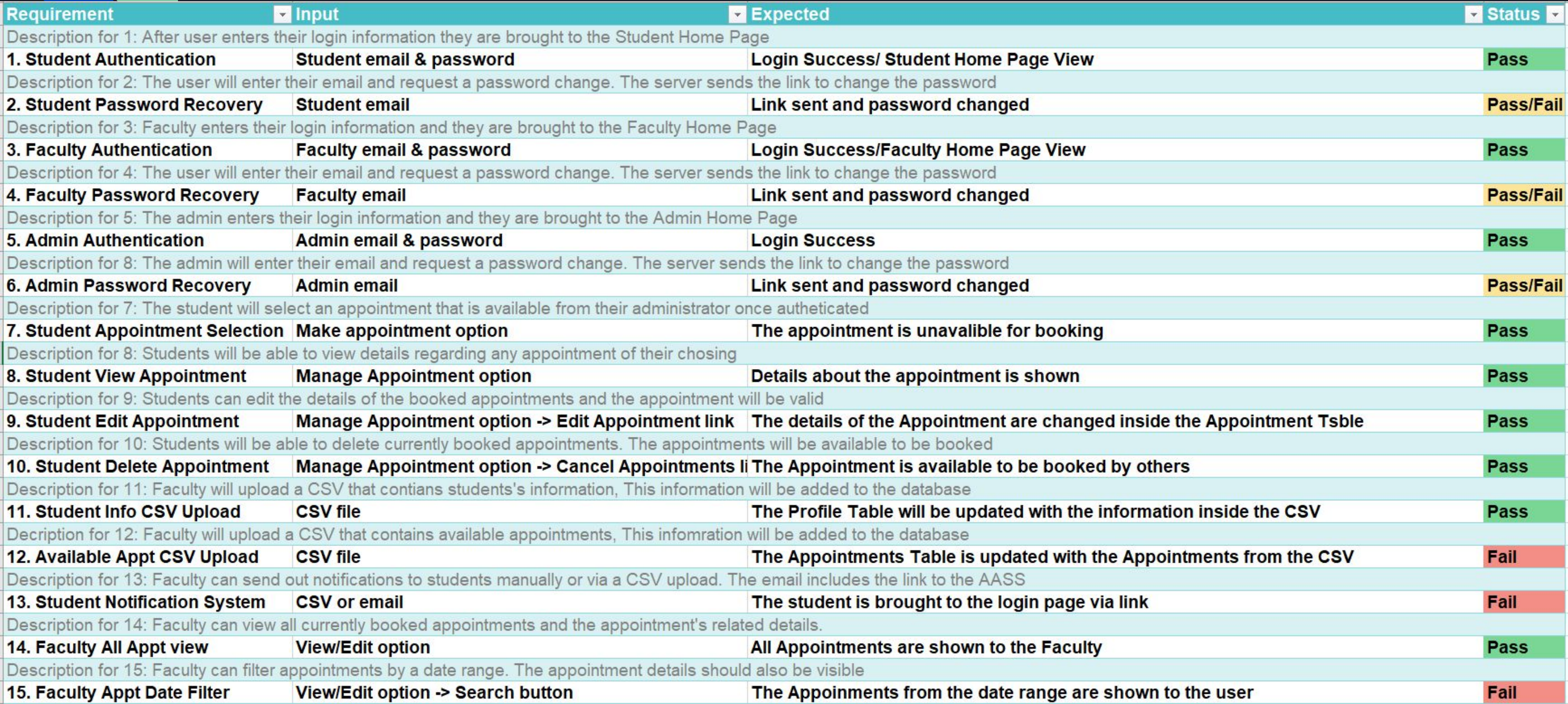
Faculty Schedule View/Filter - Not implemented

Faculty Schedule CSV Export - Not implemented

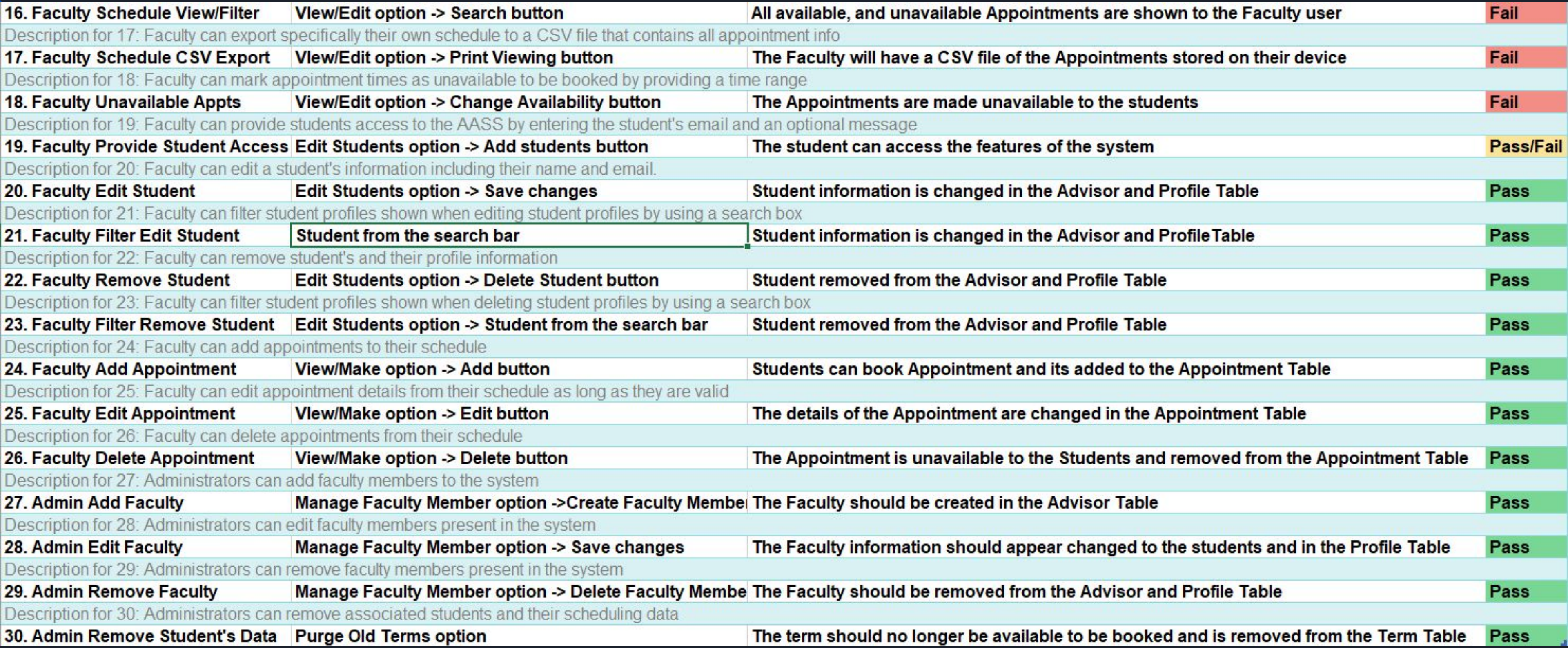
Faculty Unavailable Appts - Not implemented

Faculty Provide Student Access - Cannot add note to invitiation

Integration Testing

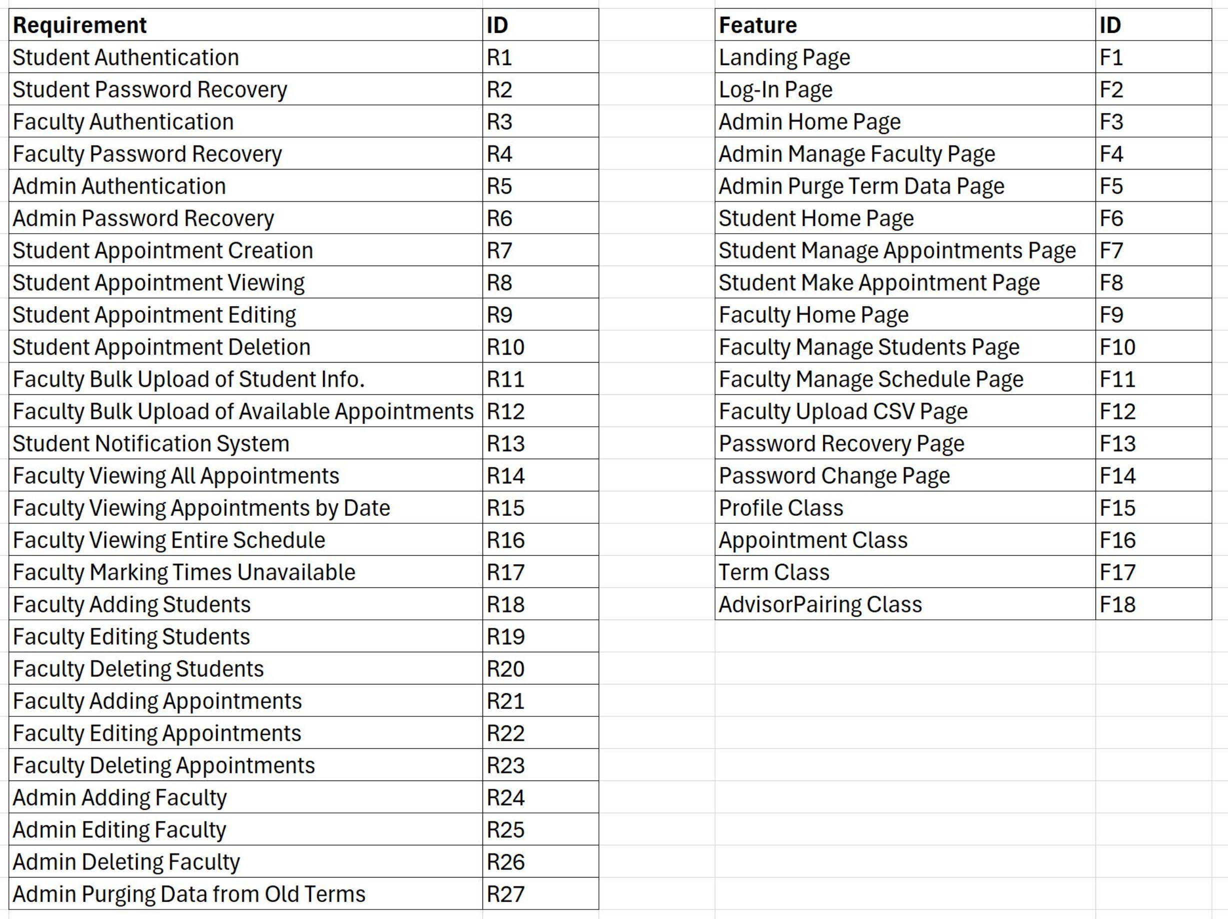


Requirement	Input	Expected	Status
Description for 1: After user enters their login information they are brought to the Student Home Page			
1. Student Authentication	Student email & password	Login Success/ Student Home Page View	Pass
Description for 2: The user will enter their email and request a password change. The server sends the link to change the password			
2. Student Password Recovery	Student email	Link sent and password changed	Pass/Fail
Description for 3: Faculty enters their login information and they are brought to the Faculty Home Page			
3. Faculty Authentication	Faculty email & password	Login Success/Faculty Home Page View	Pass
Description for 4: The user will enter their email and request a password change. The server sends the link to change the password			
4. Faculty Password Recovery	Faculty email	Link sent and password changed	Pass/Fail
Description for 5: The admin enters their login information and they are brought to the Admin Home Page			
5. Admin Authentication	Admin email & password	Login Success	Pass
Description for 8: The admin will enter their email and request a password change. The server sends the link to change the password			
6. Admin Password Recovery	Admin email	Link sent and password changed	Pass/Fail
Description for 7: The student will select an appointment that is available from their administrator once autheticated			
7. Student Appointment Selection	Make appointment option	The appointment is unavalible for booking	Pass
Description for 8: Students will be able to view details regarding any appointment of their chosing			
8. Student View Appointment	Manage Appointment option	Details about the appointment is shown	Pass
Description for 9: Students can edit the details of the booked appointments and the appointment will be valid			
9. Student Edit Appointment	Manage Appointment option -> Edit Appointment link	The details of the Appointment are changed inside the Appointment Tsble	Pass
Description for 10: Students will be able to delete currently booked appointments. The appointments will be available to be booked			
10. Student Delete Appointment	Manage Appointment option -> Cancel Appointments li	The Appointment is available to be booked by others	Pass
Description for 11: Faculty will upload a CSV that contians students's information, This information will be added to the database			
11. Student Info CSV Upload	CSV file	The Profile Table will be updated with the information inside the CSV	Pass
Decription for 12: Faculty will upload a CSV that contains available appointments, This infomration will be added to the database			
12. Available Appt CSV Upload	CSV file	The Appointments Table is updated with the Appointments from the CSV	Fail
Description for 13: Faculty can send out notifications to students manually or via a CSV upload. The email includes the link to the AASS			
13. Student Notification System	CSV or email	The student is brought to the login page via link	Fail
Description for 14: Faculty can view all currently booked appointments and the appointment's related details.			
14. Faculty All Appt view	View/Edit option	All Appointments are shown to the Faculty	Pass
Description for 15: Faculty can filter appointments by a date range. The appointment details should also be visible			
15. Faculty Appt Date Filter	View/Edit option -> Search button	The Appoinments from the date range are shown to the user	Fail



16. Faculty Schedule View/Filter	VIew/Edit option -> Search button	All available, and unavailable Appointments are shown to the Faculty user	Fail
Description for 17: Faculty can export specifically their own schedule to a CSV file that contains all appointment info			
17. Faculty Schedule CSV Export	VIew/Edit option -> Print Viewing button	The Faculty will have a CSV file of the Appointments stored on their device	Fail
Description for 18: Faculty can mark appointment times as unavailable to be booked by providing a time range			
18. Faculty Unavailable Appts	View/Edit option -> Change Availability button	The Appointments are made unavailable to the students	Fail
Description for 19: Faculty can provide students access to the AASS by entering the student's email and an optional message			
19. Faculty Provide Student Access	Edit Students option -> Add students button	The student can access the features of the system	Pass/Fail
Description for 20: Faculty can edit a student's information including their name and email.			
20. Faculty Edit Student	Edit Students option -> Save changes	Student information is changed in the Advisor and Profile Table	Pass
Description for 21: Faculty can filter student profiles shown when editing student profiles by using a search box			
21. Faculty Filter Edit Student	Student from the search bar	Student information is changed in the Advisor and ProfileTable	Pass
Description for 22: Faculty can remove student's and their profile information			
22. Faculty Remove Student	Edit Students option -> Delete Student button	Student removed from the Advisor and Profile Table	Pass
Description for 23: Faculty can filter student profiles shown when deleting student profiles by using a search box			
23. Faculty Filter Remove Student	Edit Students option -> Student from the search bar	Student removed from the Advisor and Profile Table	Pass
Description for 24: Faculty can add appointments to their schedule			
24. Faculty Add Appointment	View/Make option -> Add button	Students can book Appointment and its added to the Appointment Table	Pass
Description for 25: Faculty can edit appointment details from their schedule as long as they are valid			
25. Faculty Edit Appointment	VIew/Make option -> Edit button	The details of the Appointment are changed in the Appointment Table	Pass
Description for 26: Faculty can delete appointments from their schedule			
26. Faculty Delete Appointment	View/Make option -> Delete button	The Appointment is unavailable to the Students and removed from the Appointment Table	Pass
Description for 27: Administrators can add faculty members to the system			
27. Admin Add Faculty	Manage Faculty Member option ->Create Faculty Membe	The Faculty should be created in the Advisor Table	Pass
Description for 28: Administrators can edit faculty members present in the system			
28. Admin Edit Faculty	Manage Faculty Member option -> Save changes	The Faculty information should appear changed to the students and in the Profile Table	Pass
Description for 29: Administrators can remove faculty members present in the system			
29. Admin Remove Faculty	Manage Faculty Member option -> Delete Faculty Membe	The Faculty should be removed from the Advisor and Profile Table	Pass
Description for 30: Administrators can remove associated students and their scheduling data			
30. Admin Remove Student's Data	Purge Old Terms option	The term should no longer be available to be booked and is removed from the Term Table	Pass

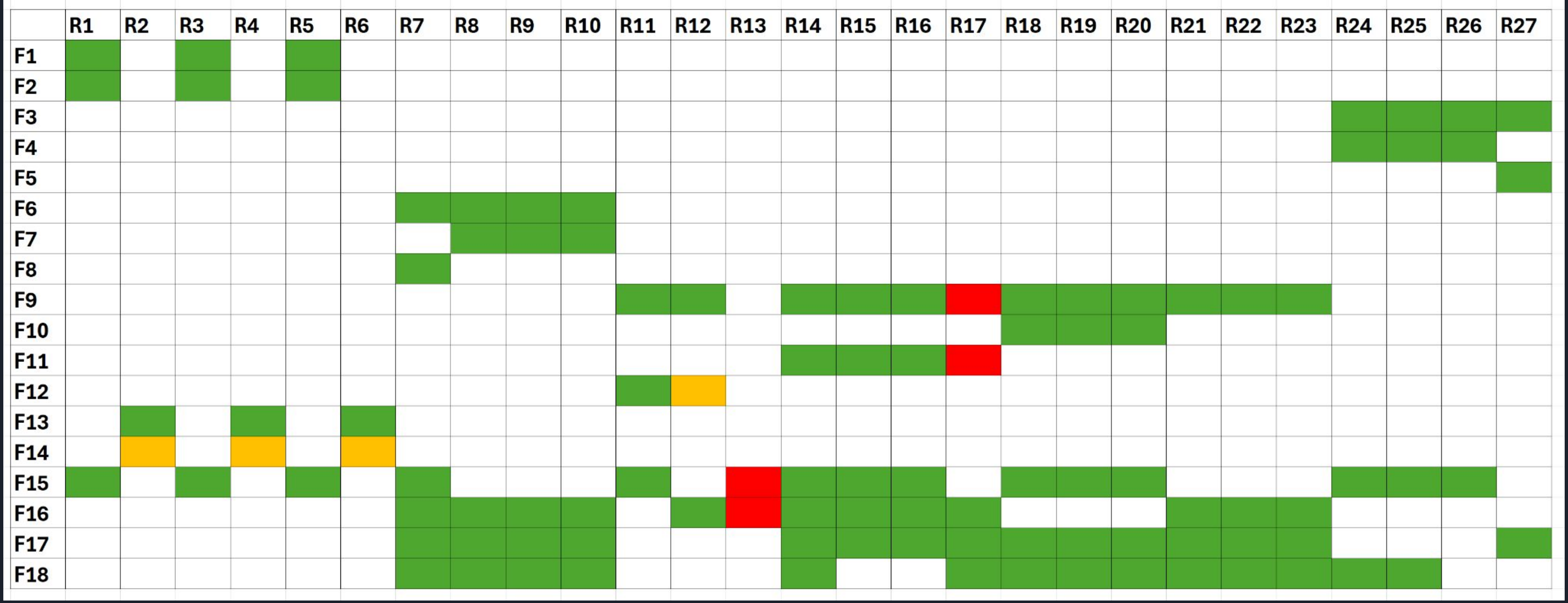
Requirements Traceability Matrix



Requirement	ID
Student Authentication	R1
Student Password Recovery	R2
Faculty Authentication	R3
Faculty Password Recovery	R4
Admin Authentication	R5
Admin Password Recovery	R6
Student Appointment Creation	R7
Student Appointment Viewing	R8
Student Appointment Editing	R9
Student Appointment Deletion	R10
Faculty Bulk Upload of Student Info.	R11
Faculty Bulk Upload of Available Appointments	R12
Student Notification System	R13
Faculty Viewing All Appointments	R14
Faculty Viewing Appointments by Date	R15
Faculty Viewing Entire Schedule	R16
Faculty Marking Times Unavailable	R17
Faculty Adding Students	R18
Faculty Editing Students	R19
Faculty Deleting Students	R20
Faculty Adding Appointments	R21
Faculty Editing Appointments	R22
Faculty Deleting Appointments	R23
Admin Adding Faculty	R24
Admin Editing Faculty	R25
Admin Deleting Faculty	R26
Admin Purging Data from Old Terms	R27

Feature	ID
Landing Page	F1
Log-In Page	F2
Admin Home Page	F3
Admin Manage Faculty Page	F4
Admin Purge Term Data Page	F5
Student Home Page	F6
Student Manage Appointments Page	F7
Student Make Appointment Page	F8
Faculty Home Page	F9
Faculty Manage Students Page	F10
Faculty Manage Schedule Page	F11
Faculty Upload CSV Page	F12
Password Recovery Page	F13
Password Change Page	F14
Profile Class	F15
Appointment Class	F16
Term Class	F17
AdvisorPairing Class	F18

Requirements Traceability Matrix Cont.



	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16	R17	R18	R19	R20	R21	R22	R23	R24	R25	R26	R27
F1	🟩		🟩		🟩																						
F2	🟩		🟩		🟩																						
F3																								🟩	🟩	🟩	🟩
F4																								🟩	🟩	🟩	
F5																											🟩
F6							🟩	🟩	🟩	🟩																	
F7								🟩	🟩	🟩																	
F8							🟩																				
F9											🟩	🟩		🟩	🟩	🟩	🟥	🟩	🟩	🟩	🟩	🟩	🟩				
F10																		🟩	🟩	🟩							
F11														🟩	🟩	🟩	🟥										
F12											🟩	🟨															
F13		🟩		🟩		🟩																					
F14		🟨		🟨		🟨																					
F15	🟩		🟩		🟩		🟩				🟩		🟥	🟩	🟩	🟩		🟩	🟩	🟩				🟩	🟩	🟩	
F16							🟩	🟩	🟩	🟩		🟩	🟥	🟩	🟩	🟩	🟩				🟩	🟩	🟩				
F17							🟩	🟩	🟩	🟩				🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩				🟩
F18							🟩	🟩	🟩	🟩				🟩			🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩		

Software Demo

Project Recap:

How it went:

- The project went well.
- Some people contributed more than others (Cooper & James MVP), but each individual's effort also contributed to the success of almost attaining a finished software product.
- Overall, the product was very close to being a MVP. It is something we can all be proud of.

Lessons Learned:

- The stakeholder's needs must be properly understood before designing and implementation.
- It takes a long time to create a quality software product. Given more time, a more fully working, non-MVP could have been completed.
- It is a team effort. Almost any contribution is valuable, and without a full team effort, progress is slowed noticeably.

Fin