Project Deliverable 3: Software Validation and Evolution

Muhammad Adnan Khan
Electrical and Computer Engineering
Concordia University

Azan Ali Shah Syed
Electrical and Computer Engineering
Concordia University

Muhammad Nofil Siddiqui
Electrical and Computer Engineering
Concordia University

Muhammad Hashim Khan
Electrical and Computer Engineering
Concordia University

Mobina Bonkar
Electrical and Computer Engineering
Concordia University

Saeideh Kalantari
Electrical and Computer Engineering
Concordia University

***Abstract*— This project focuses on the system design and development of DepartmentDirect, a Q&A platform for future students and their families during orientation events. DepartmentDirect will provide user registration, login, FAQ management, and customizable user profiles. Using agile scrum practices and C4 models, the project aims to create a user-friendly, secure, and scalable system to facilitate information sharing between prospective students and Concordia Schools' departments.**

Keywords— DepartmentDirect (DD), Software Engineering, SCRUM, Agile Modeling.

I. Introduction

An essential opportunity for prospective students and their families to become familiar with Concordia University is open house day. Future applicants, along with their family and friends, interact with faculty members at several department booths across campus during these events to get their questions answered. Although professors give thorough answers, the absence of a digital system to record and archive these exchanges means that important information is not examined in order to improve open house planning for the future, improve response guidelines, pinpoint the requirements of potential students, or come up with strategies for improving departmental applications.

The DepartmentDirect system is being developed as part of this project to try and address these issues. Future applicants' queries will be easier to collect because of this approach, which also enables departments to post updated online responses. In addition, it will examine these exchanges in order to enhance the standard of answers and pinpoint critical needs.

Future candidates will have access to news updates, event notifications, and the opportunity to ask questions and get timely responses through DepartmentDirect. The questions can cover a wide range of topics, including as using online resources, understanding course prerequisites and sequences, and using resources for career advancement.

Using DepartmentDirect, each department will be enabled to customize a version that is specific to their degree program, such as DepartmentDirect - CSE or DepartmentDirect - ECE. DepartmentDirect will allow directors of degree programs, administrative officers, and support personnel to post events and update content. Users will have flexible access to the system as it may be implemented as a chatbot, desktop software, online app, or mobile app.

This project intends to enhance the open house experience at Concordia University by leveraging digital tools to capture, store, and analyze interactions, thereby continuously improving the quality of information and services provided to prospective students and their families.

II. Core features/scenarios

A. *Scenario 1 (S1)*

Scenario 1 deals with future applicants of Concordia University who would register, manage their accounts, and customize their experience on the DepartmentDirect platform.

B. *Scenario 2 (S2)*

Scenario 2 would allow for the implementation of a system which would enable users to ask questions and receive answers from department faculties and staff, with the ability to categorize and prioritize questions.

C. *Scenario 3 (S3)*

Scenario 3 would allow departments to manage and update content, including event notifications, news updates, and resource information.

D. *Scenario 4 (S4)*

Scenario 4 would implement data analytics, providing tools for analyzing user interactions, questions, and feedback to improve the quality of answers and identify high-priority needs.

III. Epics and User Stories

A. *Epic 1: User Registration and Account Management*

1. **User Story 1.1:** Registration
   Priority: 5
   Difficulty: 3
   Story: As a future applicant, I want to create an account so that I can ask questions and receive updates.

2. **User Story1.2: Updating**
   Priority: 4
   Difficulty: 3
   Story: As a future applicant, I want to manage my profile information so that my data is accurate and up to date.

3. **User Story 1.3:** Resetting
   Priority: 4
   Difficulty: 4
   Story: As a future applicant, I want to reset my password if I forget it so that I can regain access to my account.

4. **User Story 1.4:** Administering
   Priority: 4
   Difficulty: 3
   Story: As an admin officer, I want to manage user accounts so that I can ensure the integrity of the system.

5. **User Story 1.5:** Subscribing
   Priority: 4
   Difficulty: 5
   Story: As a future applicant, I want to subscribe to notifications and event updates so that I stay informed about important events.

6. **User Story 1.6:** Deleting
   Priority: 4
   Difficulty: 3
   Story: As a future applicant, I want to delete my account if I no longer wish to use the service so that my data is removed.

B. Epic 2: Question and Answer System

1. **User Story 2.1:** Categorization
   Priority: 5
   Difficulty: 3
   Story: As a future applicant, I want to ask questions in different categories so that I can get specific information.

2. **User Story 2.2:** Responding
   Priority: 4
   Difficulty: 4
   Story: As a department staff member, I want to provide detailed answers to questions so that future applicants have the information they need.

3. **User Story 2.3:** Searching
   Priority: 4
   Difficulty: 4
   Story: As a future applicant, I want to search for previously answered questions so that I can find information quickly.

4. **User Story 2.4:** Organizing
   Priority: 4
   Difficulty: 3
   Story: As an admin officer, I want to categorize questions and answers so that they are easier to manage and access.

5. **User Story 2.5:** Notification
   Priority: 4
   Difficulty: 5
   Story: As a future applicant, I want to receive notifications when my question has been answered so that I can stay informed.

6. **User Story 2.6:** Updating
   Priority: 3
   Difficulty: 4
   Story: As a department staff member, I want to edit or update answers so that the information remains current and accurate.

C. Epic 3: Content Management

1. **User Story 3.1:** Notifications
   Priority: 3
   Difficulty: 5
   Story: As a department staff member, I want to post event notifications so that future applicants are aware of upcoming events.

2. **User Story 3.2:** Announcements
   Priority: 3
   Difficulty: 3
   Story: As a department staff member, I want to update news and announcements so that future applicants stay informed about relevant information.

3. **User Story 3.3:** Resource Management
   Priority: 4
   Difficulty: 4
   Story: As a department staff member, I want to manage resource information, such as course sequences and prerequisites, so that future applicants have accurate information.

4. **User Story 3.4:** Content Approval
   Priority: 4
   Difficulty: 3
   Story: As an admin officer, I want to approve or reject content updates so that only verified information is published.

5. **User Story 3.5:** Content Browsing
   Priority: 4
   Difficulty: 4
   Story: As a future applicant, I want to browse and access the latest content updates so that I have the most current information available.

6. **User Story 3.6:** Scheduling
   Priority: 4
   Difficulty: 4
   Story: As a department staff member, I want to schedule content postings so that information is released at appropriate times.

D. Epic 4: Data Analysis and Reporting

1. **User Story 4.1:** Analysis
   Priority: 3
   Difficulty: 5
   Story: As an admin officer, I want to analyze the questions and answers data so that I can improve future open house preparations.

2. **User Story 4.2:** Reporting
   Priority: 3
   Difficulty: 4
   Story: As a department head, I want to generate reports on user interactions so that I can identify trends and areas for improvement.

3. **User Story 4.3:** Feedback
   Priority: 4
   Difficulty: 4
   Story: As a future applicant, I want to provide feedback on the answers received so that the system can be improved.

4. **User Story 4.4:** Analytics
   Priority: 4
   Difficulty: 4
   Story: As a data analyst, I want to access detailed reports and analytics so that I can make data-driven decisions.

5. **User Story 4.5:** FAQ Identification
   Priority: 4
   Difficulty: 4
   Story: As an admin officer, I want to identify the most frequently asked questions so that I can create guidelines for standard answers.

6. **User Story 4.6:** Engagement Monitoring
   Priority:4
   Difficulty:4
   Story: As a department head, I want to monitor user engagement metrics so that I can measure the effectiveness of our content.

IV. SOFTWARE

A. GITHUB

Definition: GitHub is a web-based platform that provides hosting for software development and version control using Git. [1]

1. **Version Control:** Tracks changes in source code during software development, allowing multiple developers to collaborate on projects.
2. **Collaboration:** Provides tools for code review, pull requests, and issue tracking, facilitating collaborative development [2].
3. **Hosting:** Offers repositories for storing project code, documentation, and other files, with both public and private options.
4. **Continuous Integration:** Integrates with various CI/CD tools to automate testing and deployment.
5. **Project Management:** Includes features like project boards and wikis to manage development workflows and documentation [1] [2].

B. JIRA

Definition: JIRA is a proprietary issue tracking product developed by Atlassian that allows bug tracking and agile project management [3].

1. **Issue Tracking:** Manages and tracks issues, bugs, and tasks across development projects.
2. **Agile Project Management:** Supports agile methodologies with features like Scrum and Kanban boards, sprints, and backlog management.
3. **Workflow Customization:** Allows customization of workflows to fit specific project needs and processes.
4. **Reporting:** Provides detailed reports and dashboards to monitor progress, performance, and productivity.
5. **Integration:** Integrates with various development tools, including Bitbucket, GitHub, and Confluence, to streamline workflows [4].

C. PyCharm

Definition: PyCharm is an integrated development environment (IDE) developed by JetBrains specifically for Python programming [5].

1. **Code Editing:** Offers advanced code editing features like syntax highlighting, code completion, and refactoring.
2. **Debugging:** Provides a powerful debugger with features like breakpoints, watches, and variable inspection.
3. **Testing:** Supports various testing frameworks and offers tools for running and debugging tests.
4. **Project Management:** Manages project files and dependencies, including support for virtual environments and package management.
5. **Integration:** Integrates with version control systems like Git and various databases for seamless development workflows.

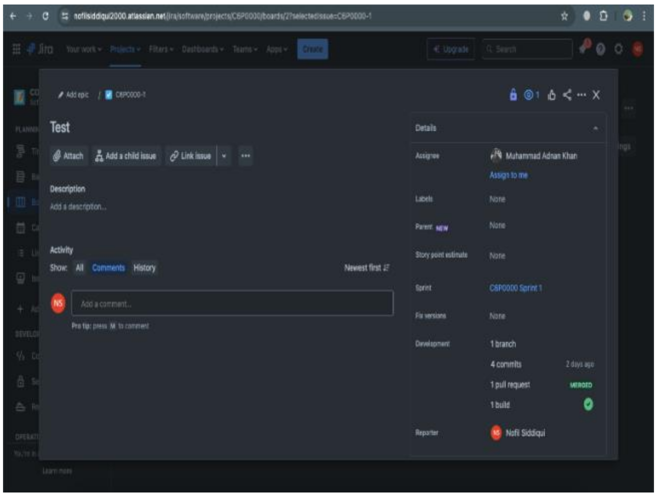
D. VSCode

Definition: Visual Studio Code (VSCode) is a free, open-source code editor developed by Microsoft. [6]

1. **Code Editing:** Offers robust code editing features including syntax highlighting, intelligent code completion, and code snippets.

2. **Extensions:** Supports a wide range of extensions to enhance functionality, including language support, debuggers, and tools for various frameworks and technologies.
3. **Debugging:** Provides an integrated debugger for multiple languages, allowing for breakpoints, step-through debugging, and variable inspection.
4. **Version Control:** Integrates with Git and other version control systems for seamless source code management.
5. **Customization:** Allows users to customize the editor through themes, keybindings, and workspace settings to suit their development preferences.

The project management tool utilized for this project was JIRA, which was seamlessly integrated with GitHub repositories. As shown in below figure



Test

Description

Add a description...

Activity

Assignee: Muhammad Adnan Khan

Labels: None

Parent NEW: None

Story point estimate: None

Sprint: C6P0000 Sprint 1

Fix versions: None

Development: 1 branch; 4 commits 2 days ago; 1 pull request MERGED; 1 build

Reporter: Nofil Siddiqui

Figure 1 Integration of Project Management tool

V. Diagrams

A. Hierarchy Diagram

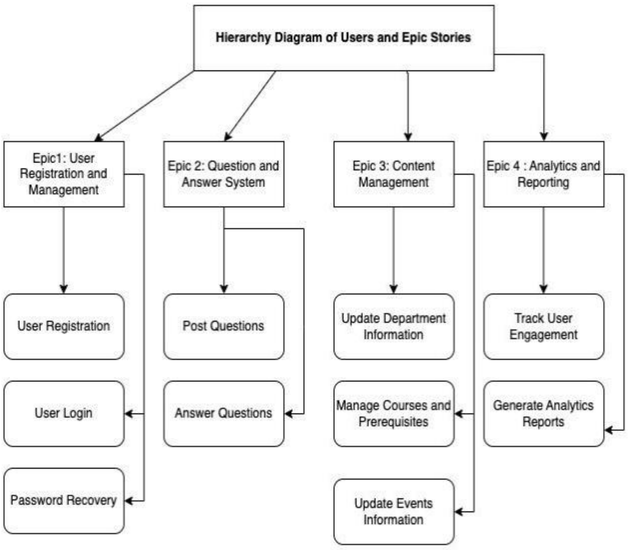

Hierarchy Diagram of Users and Epic Stories
Epic1: User Registration and Management
Epic 2: Question and Answer System
Epic 3: Content Management
Epic 4 : Analytics and Reporting
User Registration
User Login
Password Recovery
Post Questions
Answer Questions
Update Department Information
Manage Courses and Prerequisites
Update Events Information
Track User Engagement
Generate Analytics Reports

Figure 2 Hierarchy Diagram

The Hierarchy Diagram breaks down the Software System to its lowest level with the rectangles labeling the specific components or parts of a system and lines showing how they are linked to each other.

B. System Context Diagram

1) System Contex of DepartmentDirect

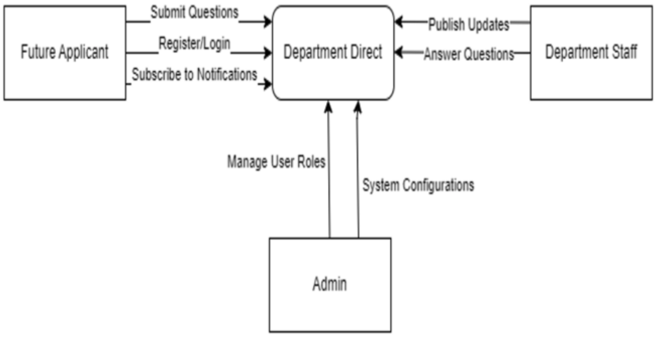


Figure 3 System Context of DepartmentDirect

The System Context diagram shows the system's environment and interactions with external entities. A System Context diagram provides a starting point, showing how the software system in scope fits into the world around it.

2) System Context of Scenario 2 (S2)

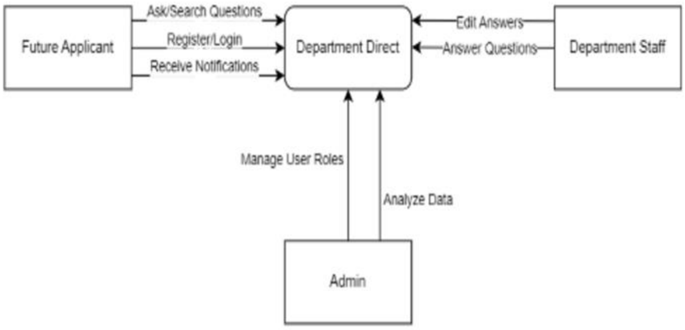


Figure 4 System Context of S2

3) System Context of Scenario 4 (S4)

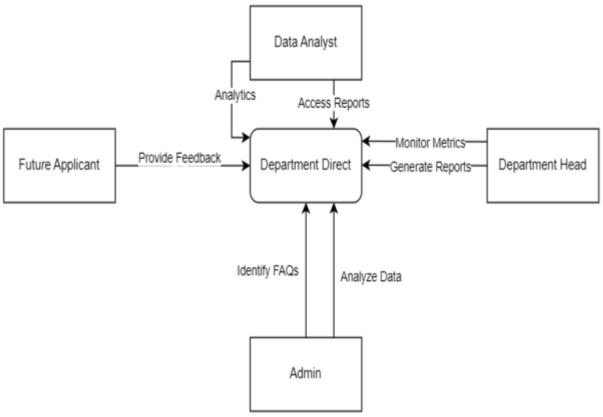


Figure 5 System Context of S4

C. Container Diagram

1) Container Diagram of DepartmentDirect

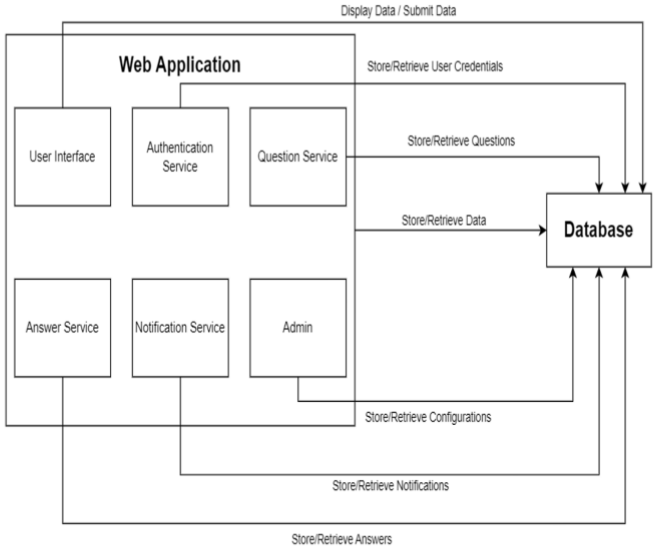


Figure 6 Container Diagram of DD

In the C4 model, a container represents an application or a data store. A container is something that needs to be running in order for the overall software system to work. A Container diagram zooms into the software system in scope, showing the high-level technical building blocks.

2) Container Diagram of Scenario 2 (S2)

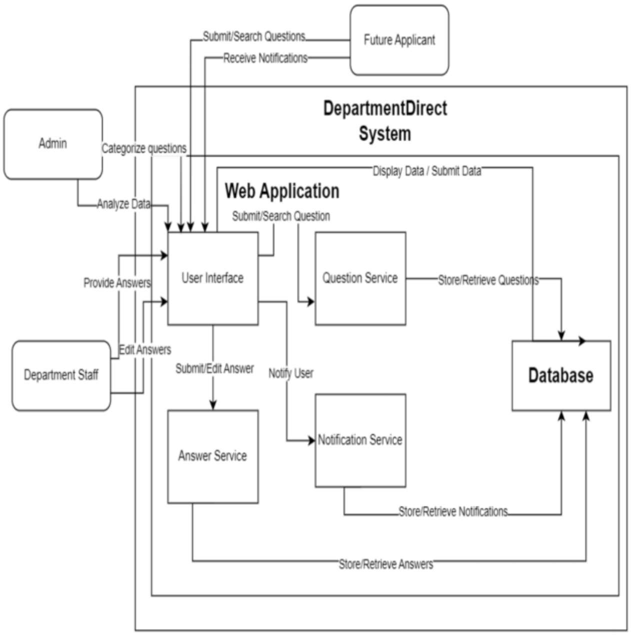


Figure 7 Container Diagram of S2

3) Container Diagram of Scenario 4 S4

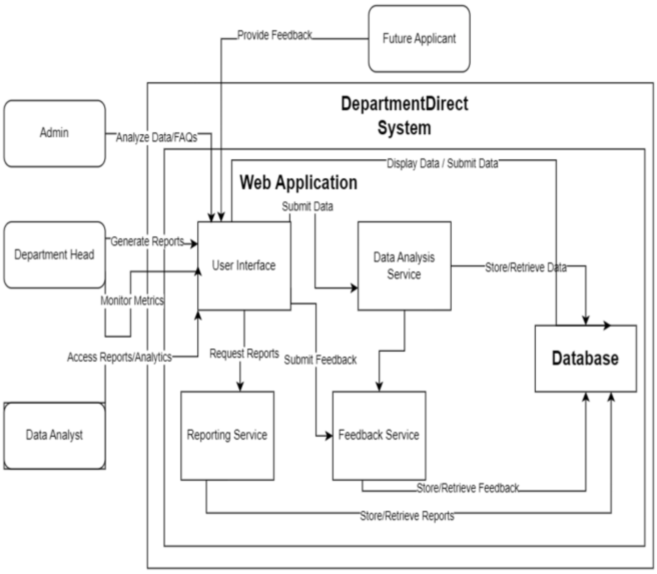


Figure 8 Container Diagram for S4

D. Component Diagram

1) Component Diagram of DepartmentDirect

`

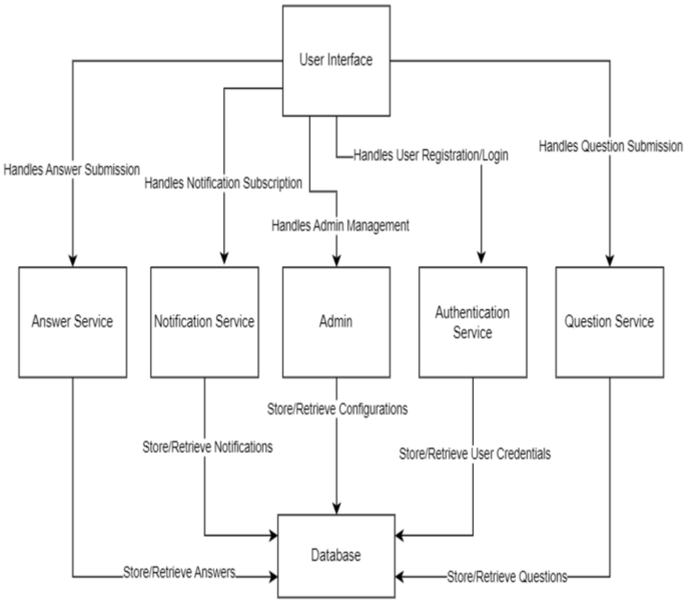

User Interface
Handles Answer Submission
Handles Notification Subscription
Handles User Registration/Login
Handles Admin Management
Handles Question Submission
Answer Service
Notification Service
Admin
Authentication Service
Question Service
Store/Retrieve Configurations
Store/Retrieve Notifications
Store/Retrieve User Credentials
Database
Store/Retrieve Answers
Store/Retrieve Questions

Figure 9 Component Diagram of DD

A Component diagram zooms into an individual container, showing the components inside it. The Component diagram shows how a container is made up of a number of "components", what each of those components are, their responsibilities and the technology/implementation details.

2) Component Diagram for Scenario 2 (S2)

`

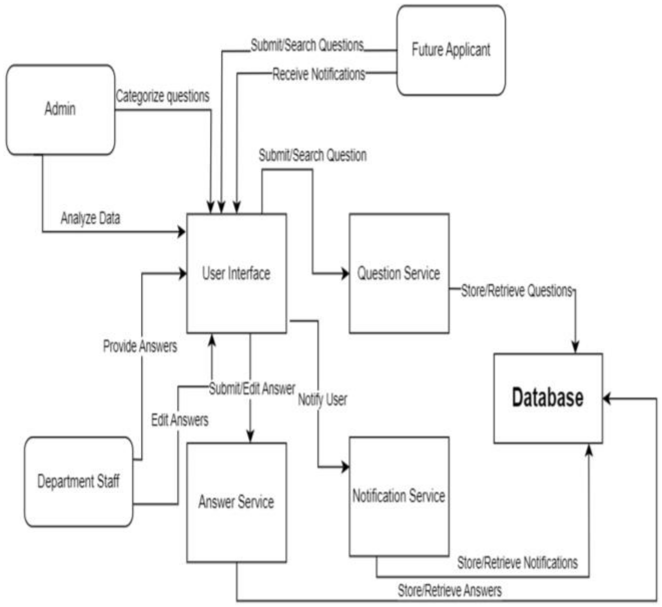


Figure 10 Component Diagram of S2

3)Component Diagram of Scenario 4 (S4)

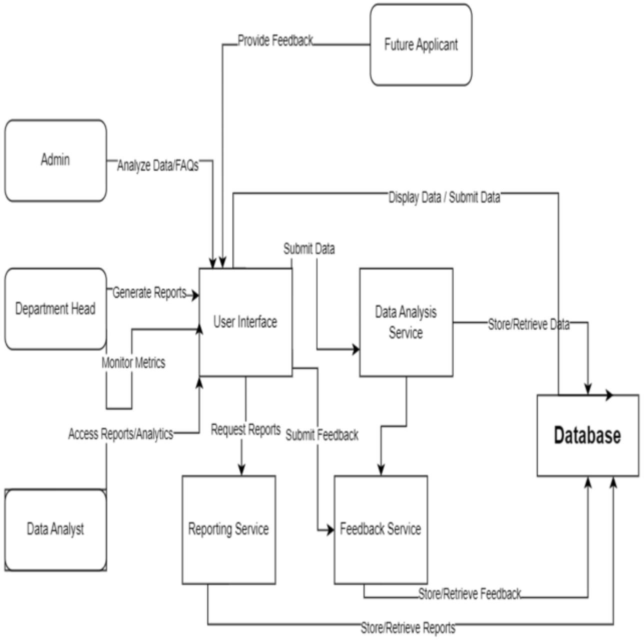


Figure 11 Component Diagram of S4

VI. ARCHITECTURE PATTERN

The system architecture includes five layers: User Interface, Application, Data, Security, and Integration. Users interact with the web application to submit questions and retrieve answers. The Application layer manages these interactions through its services. Data storage and retrieval are handled by the Data layer. The Security layer provides secure access via an Authentication Service, and the Integration layer triggers external notifications, ensuring smooth communication and reliability.

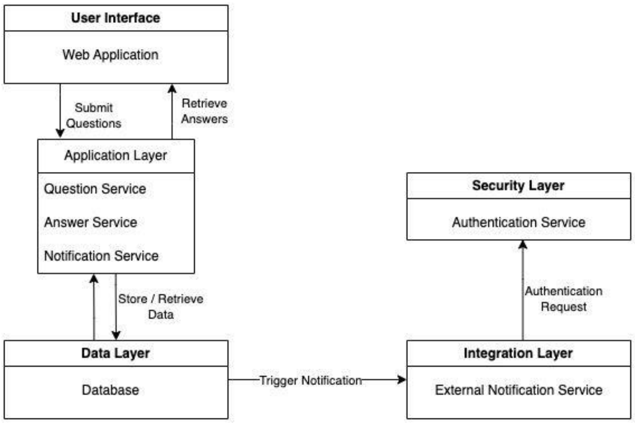


Figure 12 Architecture Design Pattern

VII. DESIGN PATTERN

Observer design pattern would be highly beneficial, particularly for the user stories that involve notifications and updates. The Observer design pattern allows an object (subject) to notify other objects (observers) about changes in its state, promoting a loose coupling between objects.

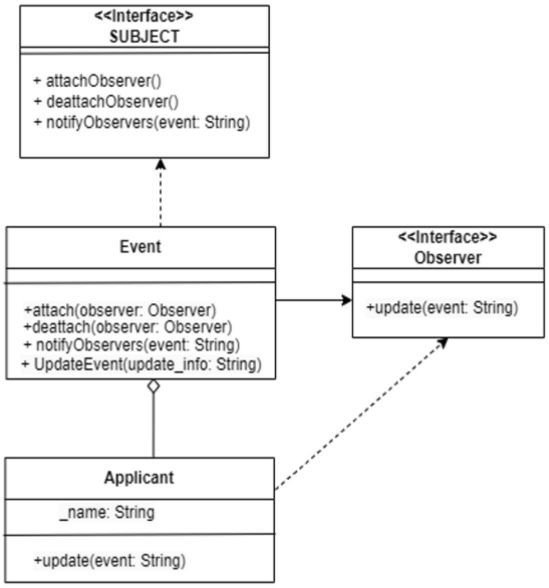

<<Interface>>
SUBJECT
+ attachObserver()
+ deattachObserver()
+ notifyObservers(event: String)
Event
+attach(observer: Observer)
+deattach(observer: Observer)
+ notifyObservers(event: String)
+ UpdateEvent(update_info: String)
<<Interface>>
Observer
+update(event: String)
Applicant
_name: String
+update(event: String)

Figure 13 Observer Design Pattern-I

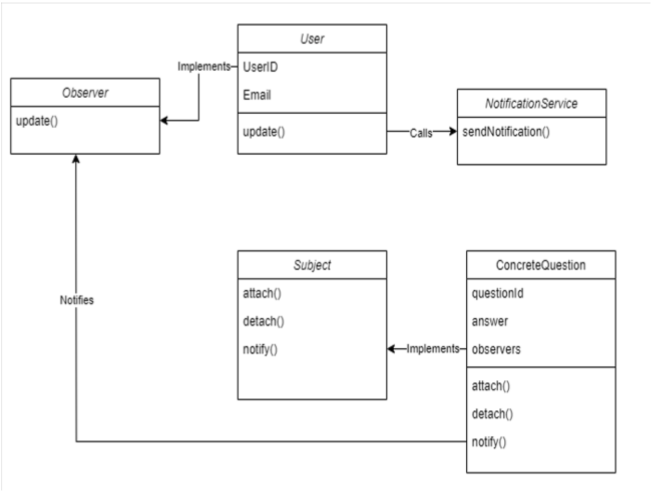


Figure 14 Design Pattern - II

VIII. C4 MODELS REVIEW

Upon evaluation, we have decided not to introduce any revisions to the C4 models at this time. The existing models accurately capture the system's architecture and support the project's requirements and objectives. Continuing with the current C4 models ensures consistency and stability in the project.

Project Management Systems

Given that no changes are being made to the C4 models, the project management system (including criteria, priorities, functionalities, time allocation, and role assignments) will remain unchanged. Here is a summary:

1. **Criteria**: Continue to focus on accuracy, completeness, and alignment with project goals.
2. **Priorities**: Maintain the existing priority order, addressing critical features and dependencies as previously planned.
3. **Functionalities**: Proceed with the development and implementation of functionalities as outlined in the project plan.
4. **Time Allocation**: Adhere to the existing timeline, ensuring timely delivery of milestones and sprints.
5. **Role Assignments**: Keep the current role assignments, leveraging the strengths and expertise of team members.

Reasons for No Change

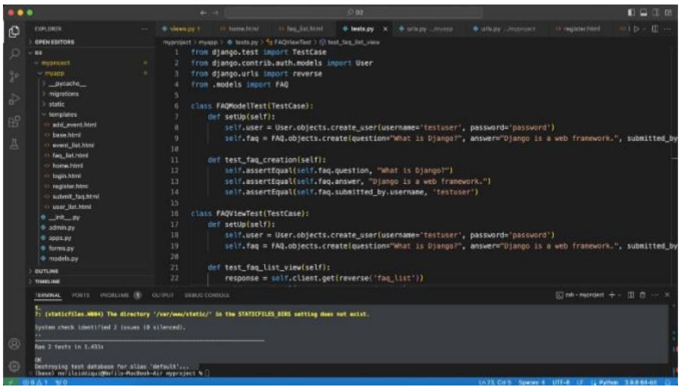
1. **Consistency**: Maintaining the current models ensures stability and avoids introducing unnecessary complexity.
2. **Accuracy**: The existing models have been reviewed and found to accurately represent the system's architecture.
3. **Efficiency**: Avoiding changes allows the team to focus on development and implementation without additional overhead from revisions.
4. **Alignment**: The current models align well with the project's goals and requirements.

In conclusion, we will continue with the existing C4 models and project management plans, as they are well-aligned with our objectives and provide a solid foundation for the DepartmentDirect project.

IX. TESTING

Unit testing is a software testing method where individual units or components of the software are tested in isolation from the rest of the application. A "unit" is the smallest testable part of any software, typically a function, method, or class.

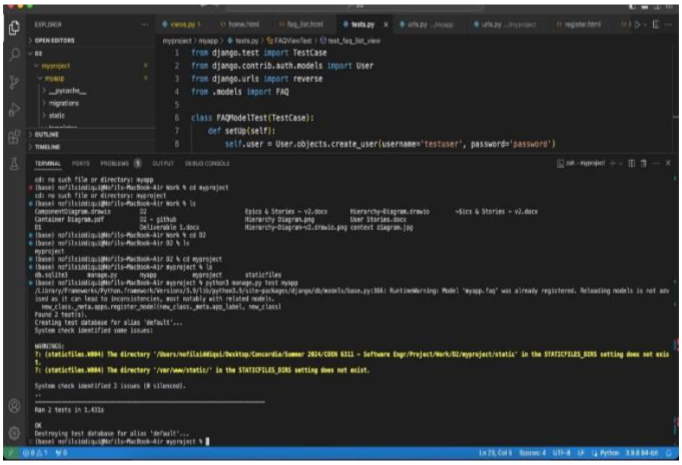
The main purpose of unit testing is to validate that each unit of the software performs as expected. It ensures that the individual components of the software work correctly in isolation.



```python
from django.test import TestCase
from django.contrib.auth.models import User
from django.urls import reverse
from .models import FAQ

class FAQModelTest(TestCase):
    def setUp(self):
        self.user = User.objects.create_user(username='testuser', password='password')
        self.faq = FAQ.objects.create(question="What is Django?", answer="Django is a web framework.", submitted_by

    def test_faq_creation(self):
        self.assertEqual(self.faq.question, "What is Django?")
        self.assertEqual(self.faq.answer, "Django is a web framework.")
        self.assertEqual(self.faq.submitted_by.username, 'testuser')

class FAQViewTest(TestCase):
    def setUp(self):
        self.user = User.objects.create_user(username='testuser', password='password')
        self.faq = FAQ.objects.create(question="What is Django?", answer="Django is a web framework.", submitted_by

    def test_faq_list_view(self):
        response = self.client.get(reverse('faq_list'))
```

```
t.
?: (staticfiles.W004) The directory '/var/www/static/' in the STATICFILES_DIRS setting does not exist.

System check identified 2 issues (0 silenced).
..
----------------------------------------------------------------------
Ran 2 tests in 1.431s

OK
Destroying test database for alias 'default'...
(base) nofilsiddiqui@Nofils-MacBook-Air myproject % 
```

Figure 15 Unit Test DepartmentDirect – I

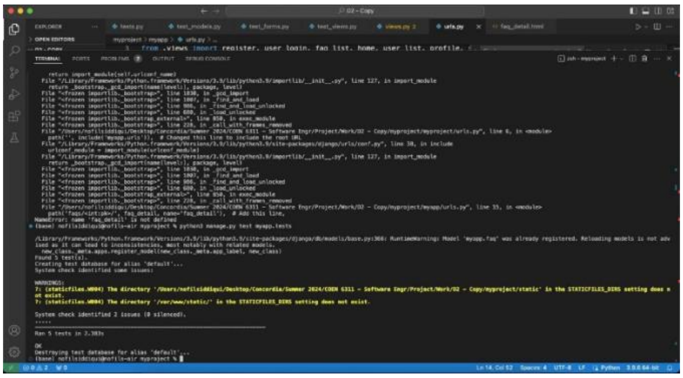


```python
from django.test import TestCase
from django.contrib.auth.models import User
from django.urls import reverse
from .models import FAQ

class FAQModelTest(TestCase):
    def setUp(self):
        self.user = User.objects.create_user(username='testuser', password='password')
```

```
cd: no such file or directory: myapp
(base) nofilsiddiqui@Nofils-MacBook-Air Work % cd myproject
cd: no such file or directory: myproject
(base) nofilsiddiqui@Nofils-MacBook-Air Work % ls
ComponentDiagram.drawio        D2                       Epics & Stories - v2.docx      Hierarchy-Diagram.drawio      ~$ics & Stories - v2.docx
Container Diagram.pdf          D2 - github              Hierarchy Diagram.png          User Stories.docx
D1                             Deliverable 1.docx       Hierarchy-Diagram-v2.drawio.png context diagram.jpg
(base) nofilsiddiqui@Nofils-MacBook-Air Work % cd D2
(base) nofilsiddiqui@Nofils-MacBook-Air D2 % ls
myproject
(base) nofilsiddiqui@Nofils-MacBook-Air D2 % cd myproject
(base) nofilsiddiqui@Nofils-MacBook-Air myproject % ls
db.sqlite3      manage.py       myapp           myproject       staticfiles
(base) nofilsiddiqui@Nofils-MacBook-Air myproject % python3 manage.py test myapp
/Library/Frameworks/Python.framework/Versions/3.9/lib/python3.9/site-packages/django/db/models/base.py:366: RuntimeWarning: Model 'myapp.faq' was already registered. Reloading models is not adv
ised as it can lead to inconsistencies, most notably with related models.
  new_class._meta.apps.register_model(new_class._meta.app_label, new_class)
Found 2 test(s).
Creating test database for alias 'default'...
System check identified some issues:

WARNINGS:
?: (staticfiles.W004) The directory '/Users/nofilsiddiqui/Desktop/Concordia/Summer 2024/COEN 6311 - Software Engr/Project/Work/D2/myproject/static' in the STATICFILES_DIRS setting does not exis
t.
?: (staticfiles.W004) The directory '/var/www/static/' in the STATICFILES_DIRS setting does not exist.

System check identified 2 issues (0 silenced).
..
----------------------------------------------------------------------
Ran 2 tests in 1.431s

OK
Destroying test database for alias 'default'...
(base) nofilsiddiqui@Nofils-MacBook-Air myproject % 
```

Figure 16 Unit Test DepartmentDirect – II

Component testing, also known as integration testing at a smaller scale, focuses on testing the interactions between multiple units or components of the software. It verifies that combined parts of the system work together as intended. The goal of component testing is to ensure that different parts of the system interact correctly and that integrated units function as expected. It tests the behavior and interactions of these combined components.

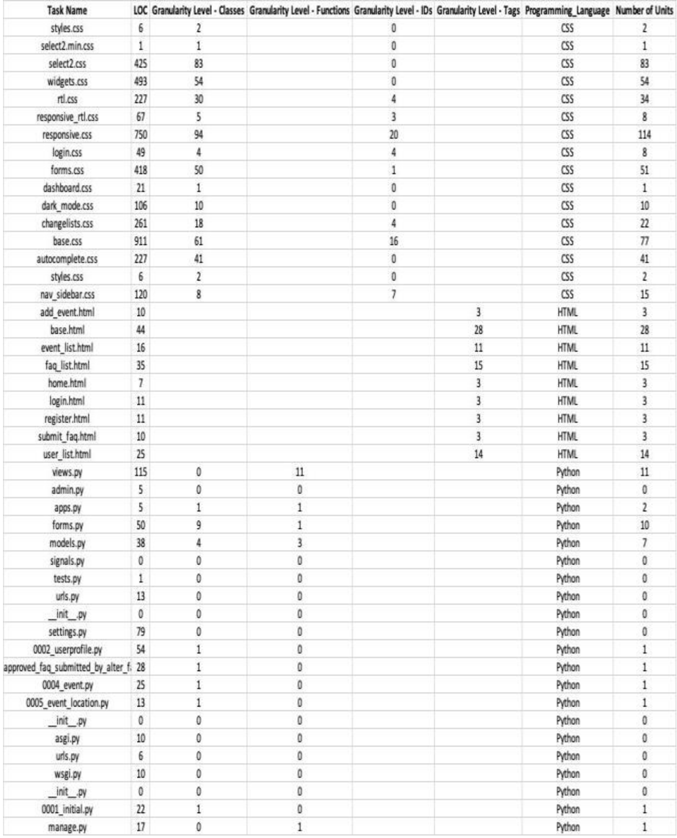


```
    return import_module(self.urlconf_name)
  File "/Library/Frameworks/Python.framework/Versions/3.9/lib/python3.9/importlib/__init__.py", line 127, in import_module
    return _bootstrap._gcd_import(name[level:], package, level)
  File "<frozen importlib._bootstrap>", line 1030, in _gcd_import
  File "<frozen importlib._bootstrap>", line 1007, in _find_and_load
  File "<frozen importlib._bootstrap>", line 986, in _find_and_load_unlocked
  File "<frozen importlib._bootstrap>", line 680, in _load_unlocked
  File "<frozen importlib._bootstrap_external>", line 850, in exec_module
  File "<frozen importlib._bootstrap>", line 228, in _call_with_frames_removed
  File "/Users/nofilsiddiqui/Desktop/Concordia/Summer 2024/COEN 6311 - Software Engr/Project/Work/D2 - Copy/myproject/myproject/urls.py", line 6, in <module>
    path('', include('myapp.urls')),  # Changed this line to include the root URL
  File "/Library/Frameworks/Python.framework/Versions/3.9/lib/python3.9/site-packages/django/urls/conf.py", line 38, in include
    urlconf_module = import_module(urlconf_module)
  File "/Library/Frameworks/Python.framework/Versions/3.9/lib/python3.9/importlib/__init__.py", line 127, in import_module
    return _bootstrap._gcd_import(name[level:], package, level)
  File "<frozen importlib._bootstrap>", line 1030, in _gcd_import
  File "<frozen importlib._bootstrap>", line 1007, in _find_and_load
  File "<frozen importlib._bootstrap>", line 986, in _find_and_load_unlocked
  File "<frozen importlib._bootstrap>", line 680, in _load_unlocked
  File "<frozen importlib._bootstrap_external>", line 850, in exec_module
  File "<frozen importlib._bootstrap>", line 228, in _call_with_frames_removed
  File "/Users/nofilsiddiqui/Desktop/Concordia/Summer 2024/COEN 6311 - Software Engr/Project/Work/D2 - Copy/myproject/myapp/urls.py", line 15, in <module>
    path('faqs/<int:pk>/', faq_detail, name='faq_detail'),  # Add this line,
NameError: name 'faq_detail' is not defined
(base) nofilsiddiqui@nofils-air myproject % python3 manage.py test myapp.tests

/Library/Frameworks/Python.framework/Versions/3.9/lib/python3.9/site-packages/django/db/models/base.py:366: RuntimeWarning: Model 'myapp.faq' was already registered. Reloading models is not adv
ised as it can lead to inconsistencies, most notably with related models.
  new_class._meta.apps.register_model(new_class._meta.app_label, new_class)
Found 5 test(s).
Creating test database for alias 'default'...
System check identified some issues:

WARNINGS:
?: (staticfiles.W004) The directory '/Users/nofilsiddiqui/Desktop/Concordia/Summer 2024/COEN 6311 - Software Engr/Project/Work/D2 - Copy/myproject/static' in the STATICFILES_DIRS setting does n
ot exist.
?: (staticfiles.W004) The directory '/var/www/static/' in the STATICFILES_DIRS setting does not exist.

System check identified 2 issues (0 silenced).
.....
----------------------------------------------------------------------
Ran 5 tests in 2.383s

OK
Destroying test database for alias 'default'...
(base) nofilsiddiqui@nofils-air myproject % 
```

Figure 17 Component Testing

X. SOFTWARE METRICS

The table below presents key software metrics gathered during the development of the DepartmentDirect system. These metrics include lines of code (LOC), granularity levels for classes, functions, IDs, and tags, as well as the programming language and number of units per file. By monitoring these metrics, we ensured high standards of code quality, timely feature delivery, and robust testing coverage, contributing to the project's overall stability and performance.



Task Name	LOC	Granularity Level - Classes	Granularity Level - Functions	Granularity Level - IDs	Granularity Level - Tags	Programming_Language	Number of Units
styles.css	6	2		0		CSS	2
select2.min.css	1	1		0		CSS	1
select2.css	425	83		0		CSS	83
widgets.css	493	54		0		CSS	54
rtl.css	227	30		4		CSS	34
responsive_rtl.css	67	5		3		CSS	8
responsive.css	750	94		20		CSS	114
login.css	49	4		4		CSS	8
forms.css	418	50		1		CSS	51
dashboard.css	21	1		0		CSS	1
dark_mode.css	106	10		0		CSS	10
changelists.css	261	18		4		CSS	22
base.css	911	61		16		CSS	77
autocomplete.css	227	41		0		CSS	41
styles.css	6	2		0		CSS	2
nav_sidebar.css	120	8		7		CSS	15
add_event.html	10				3	HTML	3
base.html	44				28	HTML	28
event_list.html	16				11	HTML	11
faq_list.html	35				15	HTML	15
home.html	7				3	HTML	3
login.html	11				3	HTML	3
register.html	11				3	HTML	3
submit_faq.html	10				3	HTML	3
user_list.html	25				14	HTML	14
views.py	115	0	11			Python	11
admin.py	5	0	0			Python	0
apps.py	5	1	1			Python	2
forms.py	50	9	1			Python	10
models.py	38	4	3			Python	7
signals.py	0	0	0			Python	0
tests.py	1	0	0			Python	0
urls.py	13	0	0			Python	0
__init__.py	0	0	0			Python	0
settings.py	79	0	0			Python	0
0002_userprofile.py	54	1	0			Python	1
approved_faq_submitted_by_alter_f	28	1	0			Python	1
0004_event.py	25	1	0			Python	1
0005_event_location.py	13	1	0			Python	1
__init__.py	0	0	0			Python	0
asgi.py	10	0	0			Python	0
urls.py	6	0	0			Python	0
wsgi.py	10	0	0			Python	0
__init__.py	0	0	0			Python	0
0001_initial.py	22	1	0			Python	1
manage.py	17	0	1			Python	1

Figure 18 Software Metric DepartmentDirect

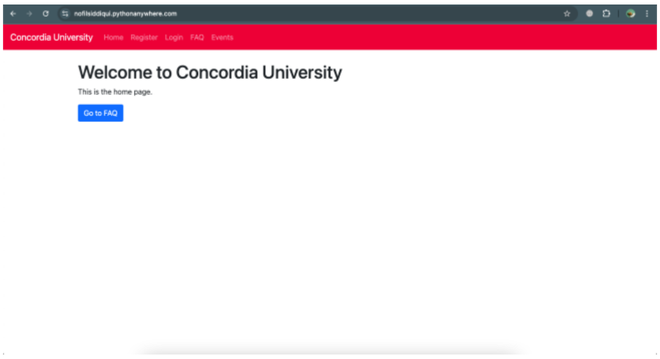
XI. Deliverable 02 Outputs

The following images showcase the final output of the DepartmentDirect system, highlighting key functionalities and user interfaces.

FAQ Page: Allows users to search for and submit frequently asked questions, enhancing user engagement and information access.

Home Page: Serves as the entry point with a welcoming interface and easy navigation to the FAQ section.

User Registration Page: Facilitates user registration by capturing essential details such as username, email, and academic interests.

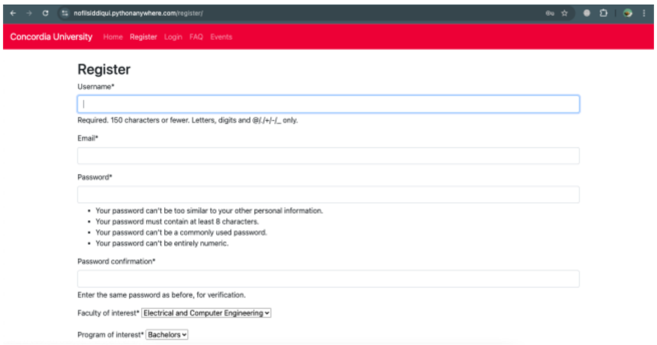


nofilsiddiqui.pythonanywhere.com

Welcome to Concordia University

This is the home page.

Figure 19 Home Page



nofilsiddiqui.pythonanywhere.com/register/

Register

Username*

Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

Email*

Password*

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

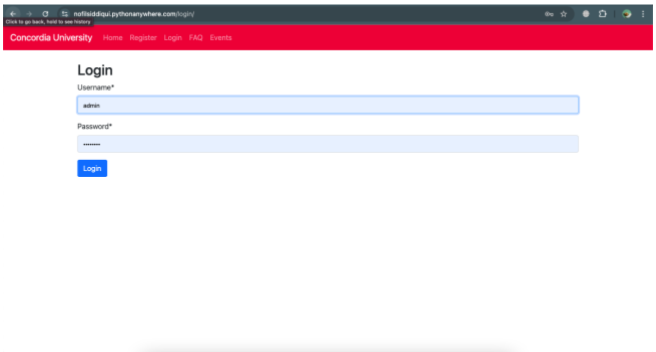
Password confirmation*

Enter the same password as before, for verification.

Faculty of interest* Electrical and Computer Engineering

Program of interest* Bachelors

Figure 20 Registration Page



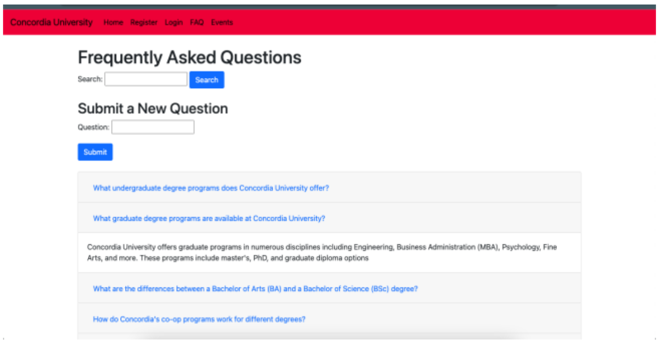
nofilsiddiqui.pythonanywhere.com/login/

Login

Username*: admin

Password*: ••••••••

Figure 21 Login Page



Frequently Asked Questions

Search:

Submit a New Question

Question:

What undergraduate degree programs does Concordia University offer?

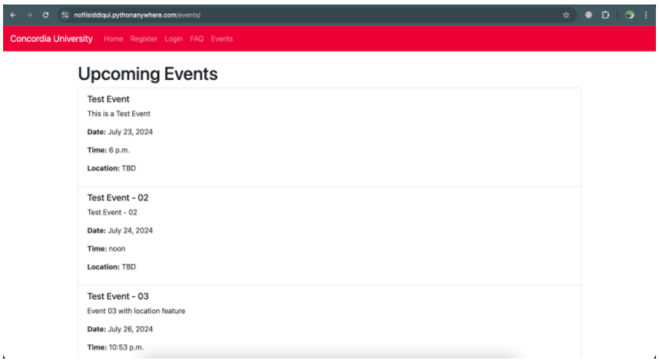
What graduate degree programs are available at Concordia University?

Concordia University offers graduate programs in numerous disciplines including Engineering, Business Administration (MBA), Psychology, Fine Arts, and more. These programs include master's, PhD, and graduate diploma options

What are the differences between a Bachelor of Arts (BA) and a Bachelor of Science (BSc) degree?

How do Concordia's co-op programs work for different degrees?

Figure 22 FAQ Page – I



nofilsiddiqui.pythonanywhere.com/events/

Upcoming Events

Test Event
This is a Test Event
Date: July 23, 2024
Time: 6 p.m.
Location: TBD

Test Event - 02
Test Event - 02
Date: July 24, 2024
Time: noon
Location: TBD

Test Event - 03
Event 03 with location feature
Date: July 26, 2024
Time: 10:53 p.m.

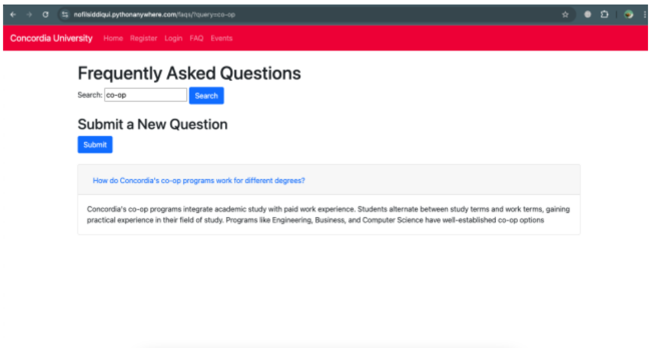
Figure 23 Event Page

XII. New Feature: Dynamic FAQ with Search Bar

The new feature aims to implement a FAQ page system by dynamically generating FAQs based on user queries and integrating a search bar for an improved user experience. To keep the FAQ list relevant and up to date, it will be regularly updated to reflect the latest trends and common queries.

Search Bar Integration adds significant value by incorporating a search bar into the FAQ page, allowing users to quickly find relevant FAQs. Real-time suggestions will be provided as users type their queries, enhancing the user experience. Additionally, the search algorithms will be improved to ensure the accuracy and relevance of search results, making it easier for users to find the information they need efficiently.

Together, these features will create a more dynamic, user-friendly FAQ system that adapts to users' needs and provides quicker, more accurate access to information.


nofilsiddiqui.pythonanywhere.com/faqs/?query=co-op

Frequently Asked Questions

Search: co-op Search

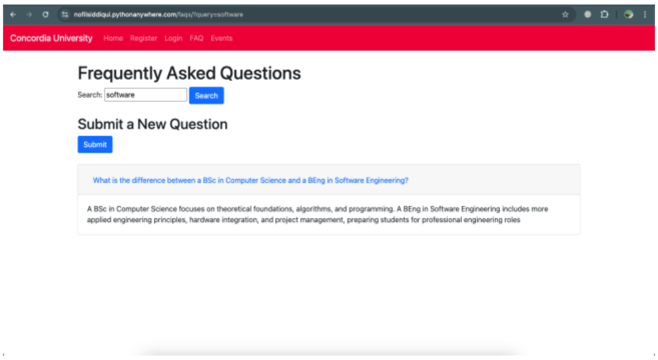
Submit a New Question

Submit

How do Concordia's co-op programs work for different degrees?

Concordia's co-op programs integrate academic study with paid work experience. Students alternate between study terms and work terms, gaining practical experience in their field of study. Programs like Engineering, Business, and Computer Science have well-established co-op options

Figure 24 Feature Implemented


nofilsiddiqui.pythonanywhere.com/faqs/?query=software

Frequently Asked Questions

Search: software Search

Submit a New Question

Submit

What is the difference between a BSc in Computer Science and a BEng in Software Engineering?

A BSc in Computer Science focuses on theoretical foundations, algorithms, and programming. A BEng in Software Engineering includes more applied engineering principles, hardware integration, and project management, preparing students for professional engineering roles

Figure 25 Search Results

XIII. Diagrams with Updated Features

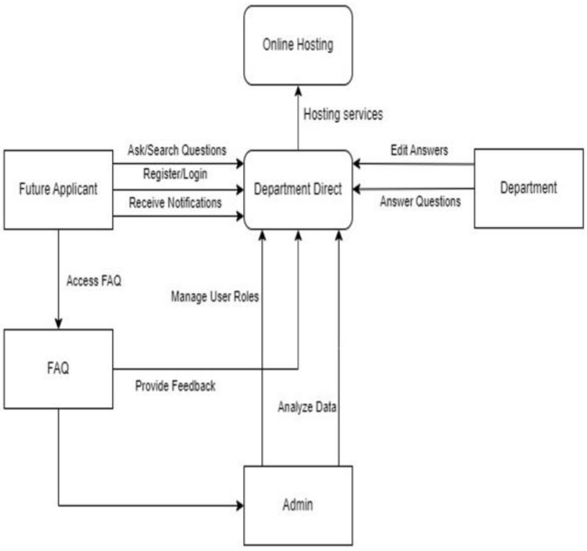


Figure 26 System Context

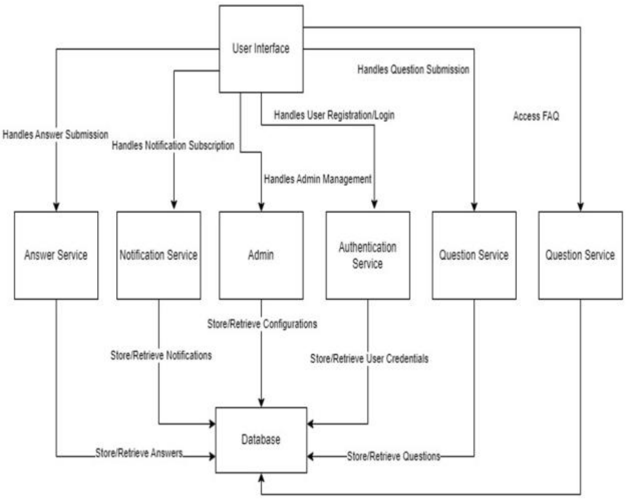

User Interface
Handles Question Submission
Handles User Registration/Login
Access FAQ
Handles Answer Submission
Handles Notification Subscription
Handles Admin Management
Answer Service
Notification Service
Admin
Authentication Service
Question Service
Question Service
Store/Retrieve Configurations
Store/Retrieve Notifications
Store/Retrieve User Credentials
Database
Store/Retrieve Answers
Store/Retrieve Questions

Figure 27 Component Diagram

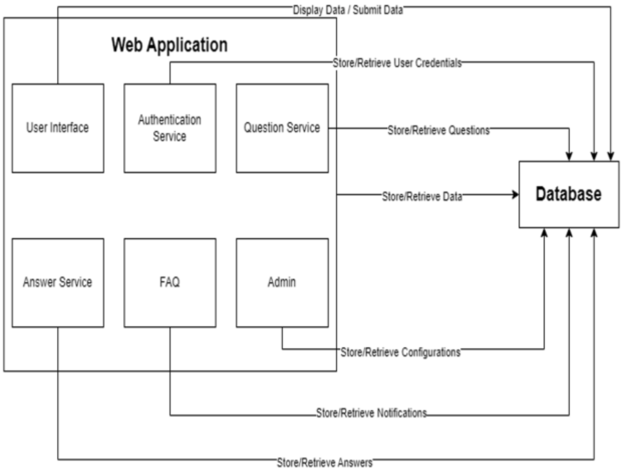


Figure 28 Container Diagram

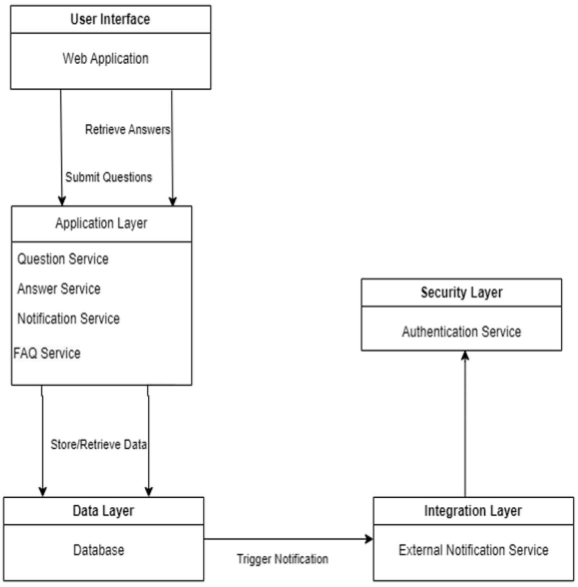


Figure 29 Architecture Diagram

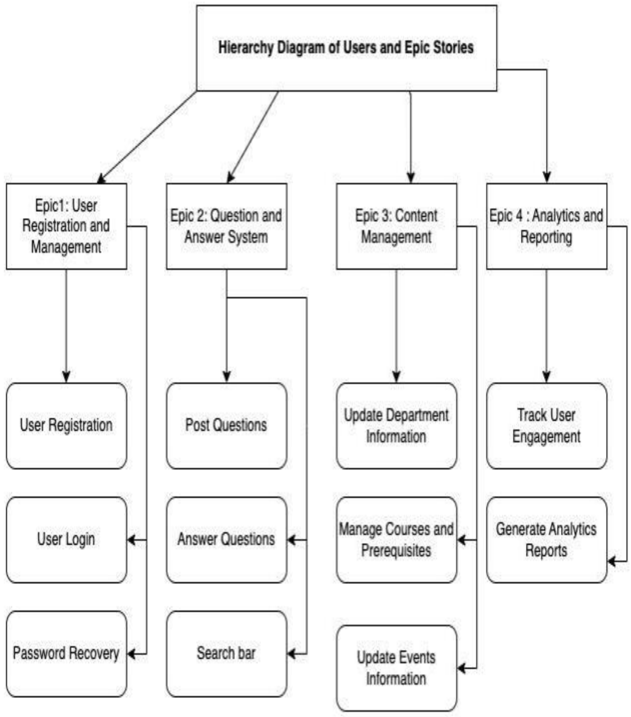

Hierarchy Diagram of Users and Epic Stories
Epic1: User Registration and Management
Epic 2: Question and Answer System
Epic 3: Content Management
Epic 4 : Analytics and Reporting
User Registration
User Login
Password Recovery
Post Questions
Answer Questions
Search bar
Update Department Information
Manage Courses and Prerequisites
Update Events Information
Track User Engagement
Generate Analytics Reports

Figure 30 Hierarchy Diagram

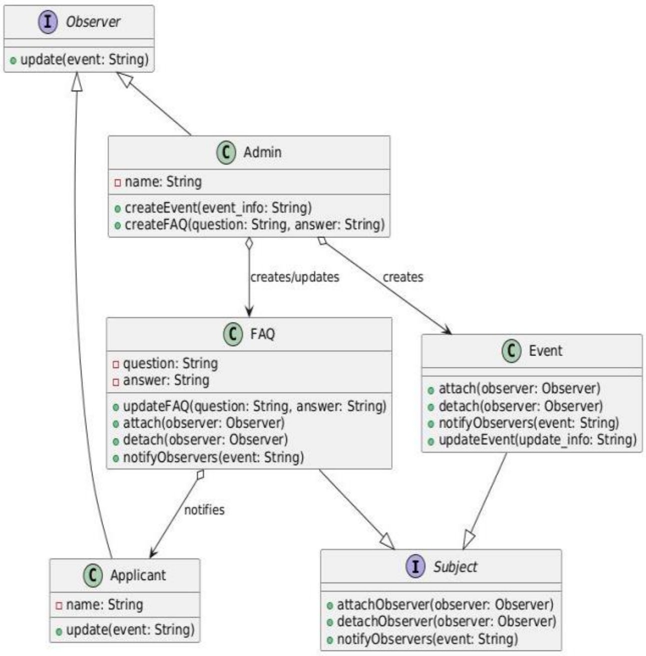

I Observer
update(event: String)
C Admin
name: String
createEvent(event_info: String)
createFAQ(question: String, answer: String)
creates/updates
creates
C FAQ
question: String
answer: String
updateFAQ(question: String, answer: String)
attach(observer: Observer)
detach(observer: Observer)
notifyObservers(event: String)
C Event
attach(observer: Observer)
detach(observer: Observer)
notifyObservers(event: String)
updateEvent(update_info: String)
notifies
C Applicant
name: String
update(event: String)
I Subject
attachObserver(observer: Observer)
detachObserver(observer: Observer)
notifyObservers(event: String)

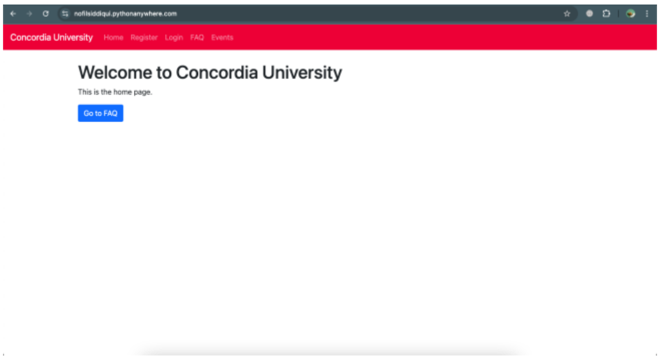
Figure 31 Observer Design Pattern

XIV. Hosting

I have utilized PythonAnywhere in conjunction with Django to host my site, leveraging the robust and scalable infrastructure that PythonAnywhere offers. This setup allows for seamless deployment and management of my Django-based project, ensuring high availability and performance. The integration of dynamic FAQ generation and a user-friendly search bar enhances the site's functionality, providing a smooth and efficient user experience. PythonAnywhere's reliable hosting environment ensures that these features operate optimally, catering to the needs of users with efficiency and ease.

URL: https://nofilsiddiqui.pythonanywhere.com/

Figure 32 QR code for URL



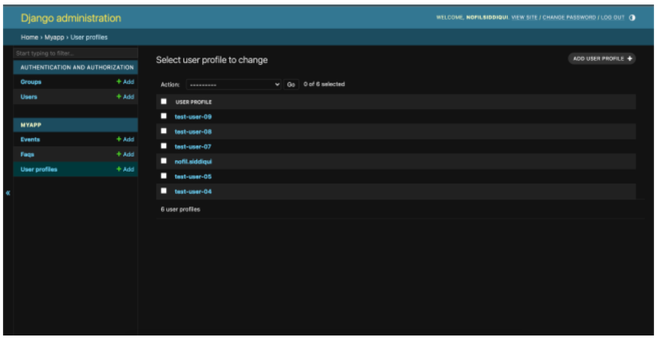
Welcome to Concordia University

This is the home page.

Figure 33 URL Hosted Site

XV. Data Base

We are utilizing Django's powerful ORM and database management capabilities to efficiently handle and store data for our site hosted on PythonAnywhere. By leveraging Django's database features, we ensure seamless integration and management of our dynamic FAQ generation system and search bar functionality. This robust setup allows for efficient logging of user queries, real-time pattern analysis, and accurate compilation of frequently asked questions. The search bar enhances user experience with real-time suggestions and improved search accuracy, all managed effortlessly through Django's reliable and scalable database framework.


Django administration
WELCOME, NOFILSIDDIQUI. VIEW SITE / CHANGE PASSWORD / LOG OUT
Home › Myapp › User profiles
Start typing to filter...
AUTHENTICATION AND AUTHORIZATION
Groups + Add
Users + Add
MYAPP
Events + Add
Faqs + Add
User profiles + Add
ADD USER PROFILE +
Action: --------- Go

Select user profile to change

0 of 6 selected

□	USER PROFILE
□	test-user-09
□	test-user-08
□	test-user-07
□	nofil.siddiqui
□	test-user-05
□	test-user-04

6 user profiles

Figure 34 Django Data Base

XVI. USABILITY AS A KEY NON-FUNCTIONAL REQUIREMENT

Usability is an important non-functional requirement for DepartmentDirect affecting the user experience. High usability means that user users can apply for subscription intuitively, locate information fast, and work with the system without unnecessary obstacles. Some of our goals relating to usability were:

- Designing an intuitive user interface.

- Providing quick access to frequently asked questions and answers.
- Providing clear navigation paths and instructions.
- Continuous improvement of the system based on users' feedback.

XVII. CONCLUSION OF THE SPRINTS

Sprint Goals Achieved: Throughout the sprints, we achieved our goals by introducing features such as a responsive design, search functionality, and easy to use dashboard.

Team Collaboration: Tools like JIRA and GitHub helped us in maintaining communication and collaboration among team members. User stories, tasks, and bugs were monitored using JIRA while GitHub facilitated version control and code review.

Continuous Improvement: After each sprint, retrospectives were done to pinpoint areas needing improvement. This approach helped us make more efficient our processes throughout further sprints.

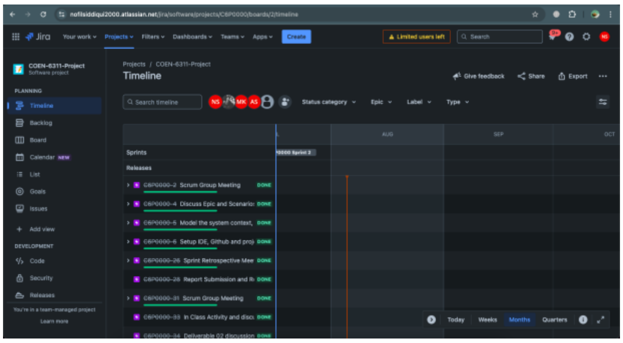
XVIII. ANALYSIS OF SCRUM PRACTICES

Daily Scrums: These meetings were conducted every day to ensure that everyone had information about what others had worked on or were held up by anything. This practice brought team coordination and efficiency.

Sprint Planning: At the beginning of every sprint there was detailed planning involving setting clear objectives and assigning tasks based on individual skills within the team. Tis was important for ensuring all the team members are working towards common objectives.

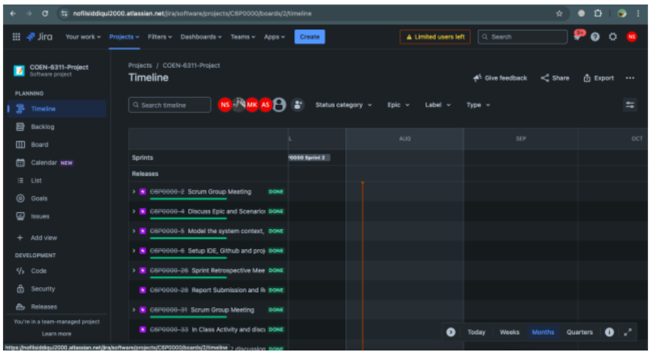
Sprint Reviews: Sprint reviews gave an overview of what was done, to get a reference and fix the problems which were found.

These practices and tools development of DeprtmentDirect remained organized and leading to a user-friendly platform.



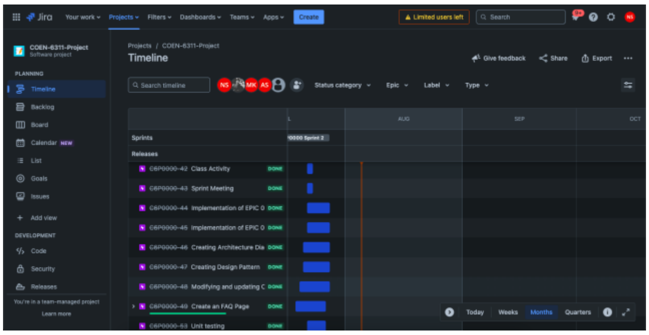
~~C6P0000-2~~ Scrum Group Meeting DONE
~~C6P0000-4~~ Discuss Epic and Scenario: DONE
~~C6P0000-5~~ Model the system context, DONE
~~C6P0000-6~~ Setup IDE, Github and proj DONE
~~C6P0000-26~~ Sprint Retrospective Mee DONE
~~C6P0000-28~~ Report Submission and R DONE
~~C6P0000-31~~ Scrum Group Meeting DONE
~~C6P0000-33~~ In Class Activity and disc DONE
~~C6P0000-34~~ Deliverable 02 discussion DONE

Figure 35 Project Management Sprint 01



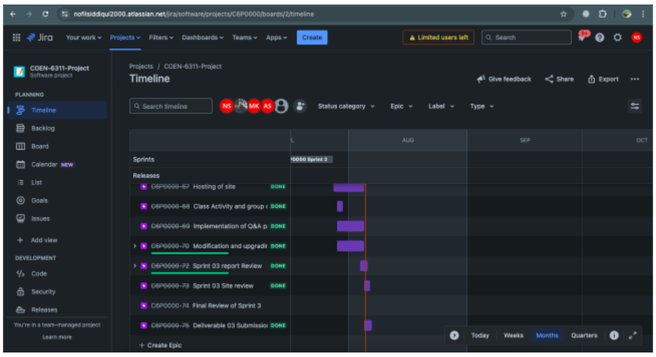
~~C6P0000-2~~ Scrum Group Meeting DONE
~~C6P0000-4~~ Discuss Epic and Scenario: DONE
~~C6P0000-5~~ Model the system context, DONE
~~C6P0000-6~~ Setup IDE, Github and proj DONE
~~C6P0000-26~~ Sprint Retrospective Mee DONE
~~C6P0000-28~~ Report Submission and R DONE
~~C6P0000-31~~ Scrum Group Meeting DONE
~~C6P0000-33~~ In Class Activity and disc DONE
2 discussion DONE

Figure 36 Project Management Sprint 2 - I



~~C6P0000-42~~ Class Activity DONE
~~C6P0000-43~~ Sprint Meeting DONE
~~C6P0000-44~~ Implementation of EPIC 0 DONE
~~C6P0000-45~~ Implementation of EPIC 0 DONE
~~C6P0000-46~~ Creating Architecture Dia DONE
~~C6P0000-47~~ Creating Design Pattern DONE
~~C6P0000-48~~ Modifying and updating C DONE
~~C6P0000-49~~ Create an FAQ Page DONE
~~C6P0000-53~~ Unit testing DONE

Figure 37 Project Management Sprint 02 – II



~~C6P0000-67~~ Hosting of site DONE
~~C6P0000-68~~ Class Activity and group c DONE
~~C6P0000-69~~ Implementation of Q&A p DONE
~~C6P0000-70~~ Modification and upgradir DONE
~~C6P0000-72~~ Sprint 03 report Review DONE
~~C6P0000-73~~ Sprint 03 Site review DONE
C6P0000-74 Final Review of Sprint 3
~~C6P0000-75~~ Deliverable 03 Submissio DONE

Figure 38 Project Management Sprint 03

XIX. CONCLUSION

In conclusion, the DepartmentDirect project aims to revolutionize the interaction between prospective students and Concordia University departments by providing a comprehensive digital platform for question answering and information dissemination. By leveraging agile methodologies like SCRUM and tools such as JIRA and GitHub, the project ensures efficient development and continuous improvement. The system not only facilitates seamless communication between future applicants and department staff but also enables data-driven decision-making through robust analytics and reporting functionalities. With its scalable architecture and customizable features, DepartmentDirect sets a new standard in enhancing open house experiences and promoting engagement within the Concordia community.

XX. References

[1] [Online]. Available: https://github.com/.

[2] [Online]. Available: https://en.wikipedia.org/wiki/GitHub.

[3] [Online]. Available: https://www.atlassian.com/software/jira.

[4] [Online]. Available: https://en.wikipedia.org/wiki/Jira_(software).

[5] [Online]. Available: https://www.jetbrains.com/pycharm/.

[6] [Online]. Available: https://code.visualstudio.com/.

[7] [Online]. Available: https://draw.io/.