COEN 6311 Software Engineering Project Deliverable - 1 (Group 2) Summer 2024

Balaji Ravindaran (40275411)
Electrical and Computer Engineering
Concordia University
Montreal, QC

Hanzhe Yang (40201362)
Electrical and Computer Engineering
Concordia University
Montreal, QC

Alexandre Panhaleux (40323505)
Aerospace Engineering
École de Technologie Supérieur
Montreal, QC

Thierry Ntieupma (40293567)
Computer and Software Engineering
Université de Montréal
Montreal, QC

***Abstract*—DepartmentDirect is a system designed to collect and analyze questions from future applicants to Concordia University, enabling departments to provide timely and detailed answers online. This platform enhances the open house experience by digitizing interactions for future analysis, helping improve answer quality and identify key needs. It supports event notifications, news updates, and real-time question responses, with customization options for each department.**

***Index Terms*—ICDE, Decoupled Q&A Architecture, API**

1. Software Specification

1.1. Functionality Description

DepartmentDirect is a comprehensive platform designed to facilitate communication and support among users, faculty, and administrators. It comprises three main modules: the User module, the Faculty module, and the Admin module.

Users, faculty, and admins can access the login page to enter their credentials, and they can create new accounts via the sign up page, with an option to sign up for a newsletter. Once logged in, users see a list of available departments and can click on any department to be redirected to a chat interface where they can view past conversations and ask new questions. Faculty members log in to view all available questions from users. They can click on any question to provide an answer and assign a category to it. The responses are then displayed in the user's chat interface, enabling direct and efficient communication between faculty and users. Admins log in to view all departments and access detailed analytics by selecting a department. The analytics page displays charts based on categories assigned by faculty, helping to analyze interaction and support metrics. Additionally, admins can send notifications to all users enrolled in the notification system, ensuring that important updates and information are effectively communicated.

1.2. Epics and User Stories

A. *Epic: Admin Management and Analytics*

User Story 001: View Department Analytics
Priority: 5 **Difficulty: 3**
As an admin, I should be able to view analytics for each department, so that I can monitor activity and performance.

User Story 002: Analyze Question Categories
Priority: 4 **Difficulty: 3**
As an admin, I should be able to analyze data based on question categories assigned by faculty, so that I can identify trends and areas needing attention.

User Story 003: Send Notifications
Priority: 5 **Difficulty: 3**
As an admin, I should be able to send notifications to all subscribed users, so that they are informed about important updates and events.

B. *Epic: Question and Answer Interaction*

User Story 001: View Available Departments
Priority: 5 **Difficulty: 2**
As a user, I should be able to see all the available departments, so that I can choose where to ask my question.

User Story 002: Ask a Question
Priority: 5 **Difficulty: 3**
As a user, I should be able to ask a question to a department, so that I can get the necessary information.

User Story 003: View Past Conversations
Priority: 4 **Difficulty: 3**
As a user, I should be able to see my past conversations, so that I can review previous answers and follow up if necessary.

User Story 004: Chat with Faculty
Priority: 5 **Difficulty: 3**
As a user, I should be able to chat with faculty members, so that I can get detailed answers to my questions.

C. *Epic: Faculty Management and Interaction*

User Story 001: View All Questions
Priority: 5 **Difficulty: 2**
As a faculty member, I should be able to view all questions asked by users, so that I can choose which ones to answer.

User Story 002: Answer Questions
Priority: 5 **Difficulty: 3**
As a faculty member, I should be able to answer questions asked by users, so that I can provide them with the necessary information.

User Story 003: Assign Categories to Questions
Priority: 4 **Difficulty: 3**
As a faculty member, I should be able to assign categories to questions, so that similar questions are grouped together for easier reference and analysis.

D. *Epic: User Account Management*

User Story 001: Registration
Priority: 5 **Difficulty: 2**
As a user, I should be able to create my own account easily, so that I can start using this application immediately by providing necessary information (name, password, email, phone number, address, etc.).

User Story 002: Log In
Priority: 5 **Difficulty: 2**
As a user, I should be able to log in easily, so that I can access the functions of this application.

User Story 003: Password Reset
Priority: 4 **Difficulty: 3**
As a user, I should be able to reset my password if I forget it, so that I can regain access to my account.

User Story 004: Manage Notification Preferences
Priority: 4 **Difficulty: 3**
As a user, I should be able to manage my notification preferences, so that I only receive relevant updates.

1.2.1. Hierarchy Diagram of Epics

A. *Epic 1: Admin Management and Analytics*

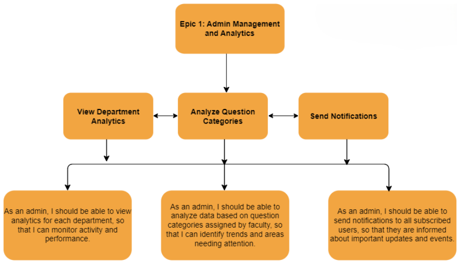

Epic 1: Admin Management and Analytics
View Department Analytics
Analyze Question Categories
Send Notifications
As an admin, I should be able to view analytics for each department, so that I can monitor activity and performance.
As an admin, I should be able to analyze data based on question categories assigned by faculty, so that I can identify trends and areas needing attention.
As an admin, I should be able to send notifications to all subscribed users, so that they are informed about important updates and events.

Fig. 1. Hierarchy Diagram for Epic 1

B. *Epic 2: Question and Answer Interaction*

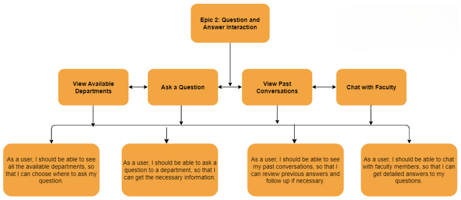

Epic 2: Question and Answer Interaction
View Available Departments
Ask a Question
View Past Conversations
Chat with Faculty
As a user, I should be able to see all the available departments, so that I can choose where to ask my question.
As a user, I should be able to ask a question to a department, so that I can get the necessary information.
As a user, I should be able to see my past conversations, so that I can review previous answers and follow up if necessary.
As a user, I should be able to chat with faculty members, so that I can get detailed answers to my questions.

Fig. 2. Hierarchy Diagram for Epic 2

C. *Epic 3: Faculty Management and Interaction*

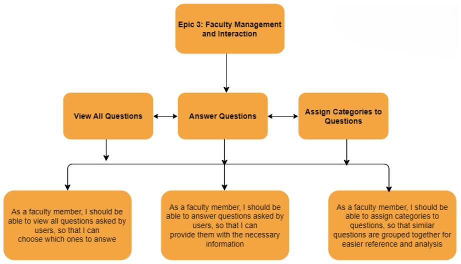

Epic 3: Faculty Management and Interaction
View All Questions
Answer Questions
Assign Categories to Questions
As a faculty member, I should be able to view all questions asked by users, so that I can choose which ones to answe
As a faculty member, I should be able to answer questions asked by users, so that I can provide them with the necessary information
As a faculty member, I should be able to assign categories to questions, so that similar questions are grouped together for easier reference and analysis

Fig. 3. Hierarchy Diagram for Epic 3

D. *Epic 4: User Account Management*

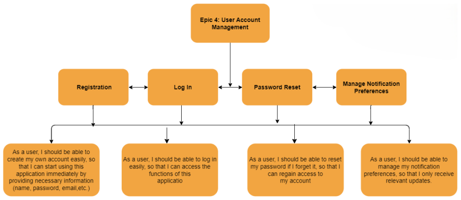

Epic 4: User Account Management
Registration
Log In
Password Reset
Manage Notification Preferences
As a user, I should be able to create my own account easily, so that I can start using this application immediately by providing necessary information (name, password, email,etc.)
As a user, I should be able to log in easily, so that I can access the functions of this applicatio
As a user, I should be able to reset my password if I forget it, so that I can regain access to my account
As a user, I should be able to manage my notification preferences, so that I only receive relevant updates.

Fig. 4. Hierarchy Diagram for Epic 4

1.3. System Context, Container and Components for Epic 2&4

A. *Epic 2: System Context*

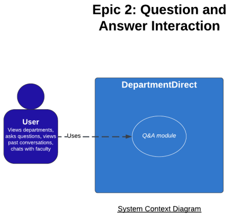

Epic 2: Question and Answer Interaction
User
Views departments, asks questions, views past conversations, chats with faculty
Uses
DepartmentDirect
Q&A module
System Context Diagram

Fig. 5. System Context Diagram for Epic 2

B. *Epic 2: Container*

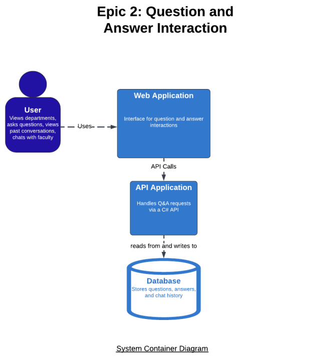

Epic 2: Question and Answer Interaction
User
Views departments, asks questions, views past conversations, chats with faculty
Uses
Web Application
Interface for question and answer interactions
API Calls
API Application
Handles Q&A requests via a C# API
reads from and writes to
Database
Stores questions, answers, and chat history
System Container Diagram

Fig. 6. Container Diagram for Epic 2

C. *Epic 2: Components*

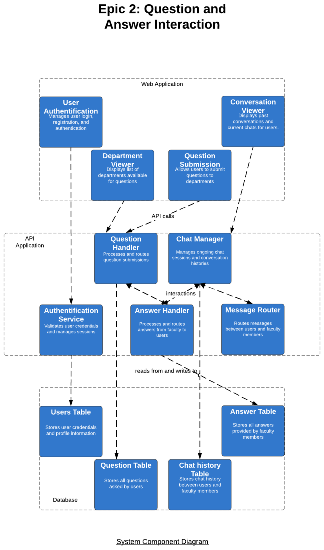

Epic 2: Question and Answer Interaction
Web Application
User Authentification
Manages user login, registration, and authentication
Conversation Viewer
Displays past conversations and current chats for users.
Department Viewer
Displays list of departments available for questions
Question Submission
Allows users to submit questions to departments
API calls
API Application
Question Handler
Processes and routes question submissions
Chat Manager
Manages ongoing chat sessions and conversation histories
interactions
Authentification Service
Validates user credentials and manages sessions
Answer Handler
Processes and routes answers from faculty to users
Message Router
Routes messages between users and faculty members
reads from and writes to
Users Table
Stores user credentials and profile information
Answer Table
Stores all answers provided by faculty members
Question Table
Stores all questions asked by users
Chat history Table
Stores chat history between users and faculty members
Database
System Component Diagram

Fig. 7. Component Diagram for Epic 2

D. *Epic 4: System Context*

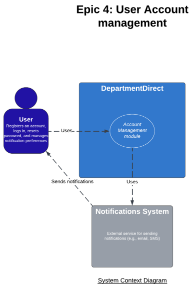

Epic 4: User Account management
User
Registers an account, logs in, resets password, and manages notification preferences
Uses
DepartmentDirect
Account Management module
Sends notifications
Uses
Notifications System
External service for sending notifications (e.g., email, SMS)
System Context Diagram

Fig. 8. System Context Diagram for Epic 4

E. *Epic 4: Container*

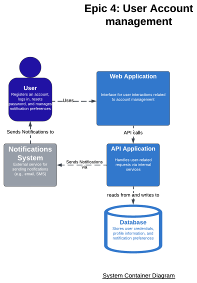

Epic 4: User Account management
User
Registers an account, logs in, resets password, and manages notification preferences
Uses
Web Application
Interface for user interactions related to account management
Sends Notifications to
API calls
Notifications System
External service for sending notifications (e.g., email, SMS)
Sends Notifications via
API Application
Handles user-related requests via internal services
reads from and writes to
Database
Stores user credentials, profile information, and notification preferences
System Container Diagram

Fig. 9. Container Diagram for Epic 4

F. *Epic 4: Components*

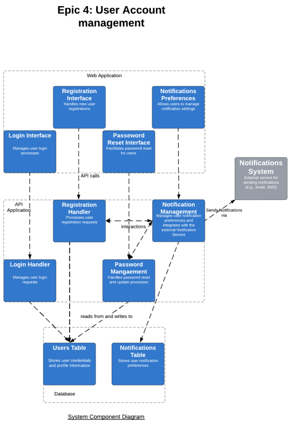

Epic 4: User Account management
Web Application
Registration Interface
Handles new user registrations
Notifications Preferences
Allows users to manage notification settings
Login Interface
Manages user login processes
Passeword Reset Interface
Facilitates password reset for users.
API calls
Notifications System
External service for sending notifications (e.g., email, SMS)
API Application
Registration Handler
Processes user registration requests
Notification Management
Manages user notification preferences and integrates with the external Notification Service
Sends Notifications via
interactions
Login Handler
Manages user login requests
Password Mangaement
Handles password reset and update processes
reads from and writes to
Users Table
Stores user credentials and profile information
Notifications Table
Stores user notification preferences
Database
System Component Diagram

Fig. 10. Component Diagram for Epic 4

1.4. Working Environment

1.4.1. Integrated Development and SQL Management Environment

A. Integrated Development Environment (IDE)

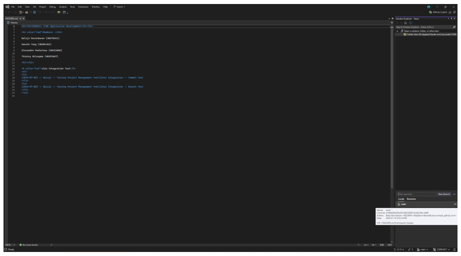


Fig. 11. Microsoft Visual Studio

B. SQL Management Environment

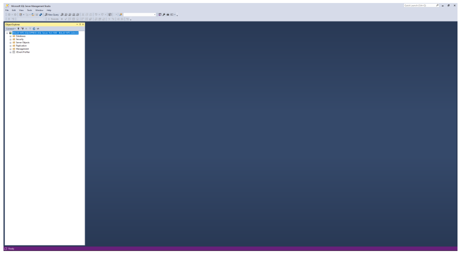
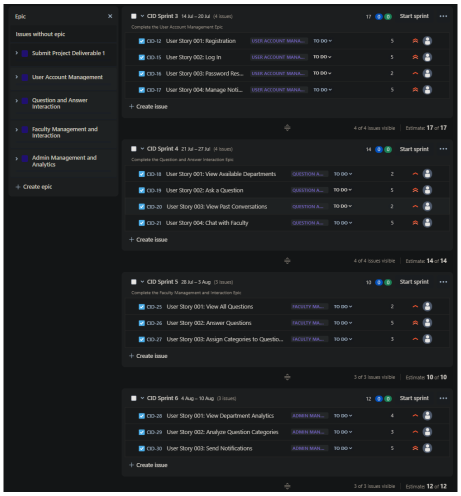


Fig. 12. Microsoft SQL Server Management Studio

1.4.2. Individual Issues For User Stories



Epic
Issues without epic
Submit Project Deliverable 1
User Account Management
Question and Answer Interaction
Faculty Management and Interaction
Admin Management and Analytics
Create epic

CID Sprint 3 14 Jul – 20 Jul (4 issues) 17 0 0 Start sprint

Complete the User Account Management Epic

CID-12	User Story 001: Registration	USER ACCOUNT MANA...	TO DO	5
CID-15	User Story 002: Log In	USER ACCOUNT MANA...	TO DO	5
CID-16	User Story 003: Password Res...	USER ACCOUNT MANA...	TO DO	2
CID-17	User Story 004: Manage Noti...	USER ACCOUNT MANA...	TO DO	5

Create issue

4 of 4 issues visible | Estimate: 17 of 17

CID Sprint 4 21 Jul – 27 Jul (4 issues) 14 0 0 Start sprint

Complete the Question and Answer Interaction Epic

CID-18	User Story 001: View Available Departments	QUESTION A...	TO DO	2
CID-19	User Story 002: Ask a Question	QUESTION A...	TO DO	5
CID-20	User Story 003: View Past Conversations	QUESTION A...	TO DO	2
CID-21	User Story 004: Chat with Faculty	QUESTION A...	TO DO	5

Create issue

4 of 4 issues visible | Estimate: 14 of 14

CID Sprint 5 28 Jul – 3 Aug (3 issues) 10 0 0 Start sprint

Complete the Faculty Management and Interaction Epic

CID-25	User Story 001: View All Questions	FACULTY MA...	TO DO	2
CID-26	User Story 002: Answer Questions	FACULTY MA...	TO DO	5
CID-27	User Story 003: Assign Categories to Questio...	FACULTY MA...	TO DO	3

Create issue

3 of 3 issues visible | Estimate: 10 of 10

CID Sprint 6 4 Aug – 10 Aug (3 issues) 12 0 0 Start sprint

CID-28	User Story 001: View Department Analytics	ADMIN MAN...	TO DO	4
CID-29	User Story 002: Analyze Question Categories	ADMIN MAN...	TO DO	3
CID-30	User Story 003: Send Notifications	ADMIN MAN...	TO DO	5

Create issue

3 of 3 issues visible | Estimate: 12 of 12

Fig. 13. Individual Issues For User Stories Segmented by Epic per Sprint

1.5. Knowledge Base on Scrum, C4 models, and User Stories

A. Team's Knowledge on Scrum

Our team has developed knowledge and practical understanding of Scrum. We have studied and practiced the core principles of Scrum, an agile framework for managing software projects. Our knowledge covers:

1) Scrum Roles:

- **Scrum Master:** Facilitates Scrum practices, helps the team remove obstacles, and organizes meetings.
- **Product Owner:** Manages the product backlog, prioritizes tasks, and ensures the team delivers value.
- **Development Team:** Works together to deliver increments of potentially shippable products.

2) Scrum Events:

- **Sprint Planning:** Setting sprint goals and selecting backlog items for each sprint.
- **Daily Stand-ups:** Daily meetings to sync activities and identify roadblocks.
- **Sprint Review:** Presenting completed work to stakeholders and gathering feedback.
- **Sprint Retrospective:** Reflecting on performance at the end of each sprint to identify areas for improvement.

3) Scrum Artifacts:

- **Product Backlog:** Maintaining a list of all necessary features and tasks.
- **Sprint Backlog:** Creating a list of tasks to be completed during each sprint.
- **Increment:** Delivering usable and valuable increments at the end of each sprint.

B. Team's Knowledge on C4 Models

Our team is learning to use the C4 model for visualizing software architecture. We focus on:

1) Context Diagrams: Creating high-level diagrams to show the system and its interactions with external entities.

2) Container Diagrams: Outlining the high-level structure of our software architecture, showing how different parts of the system interact.

3) Component Diagrams: Zooming into individual containers to understand the major building blocks and their interactions.

4) Code Diagrams: Exploring detailed views of the code to understand classes and their relationships, helping us grasp the finer details of implementation.

Using the C4 model helps us ensure that our architecture is understandable to all team members and stakeholders, facilitating better design decisions and communication.

C. Team's Knowledge on User Stories

Our team has focused on mastering user stories to capture and manage requirements effectively. Our understanding includes:

1) Structure of User Stories: We follow the format: ”As a [type of user], I want [an action] so that [a benefit/a value].” Example: ”As a user, I want to reset my password so that I can regain access to my account if I forget my password.”

2) Acceptance Criteria: Defining specific conditions that must be met for a user story to be considered complete. This helps us ensure the story meets the user's needs.

3) Story Mapping and Prioritization: Visualizing and organizing user stories to plan our releases and iterations. Prioritizing user stories based on their value to the user and the effort required.

Using user stories helps us stay user-focused and deliver increments of value in every sprint, aligning closely with our stakeholders' needs and expectations.

1.6. Conclusion

Our team has made solid progress in building a foundation in Scrum, C4 models, and user stories. We are committed to continuous learning and improvement, ensuring the successful delivery of our project.

References

[1] Dr. Yan Liu, *Lecture Notes*, Summer 2024.
[2] Jira Documentation, *Atlassian Support*, 6.4x.
[3] Ian Sommerville, *Software Engineering*, 10th Edition.