

**CS673 Software Engineering**  
**Team 1 : BU Academic Navigator (BUAN)**  
**Meeting Minutes**

All meeting minutes are kept in this single document. The latest meeting minutes should be at the beginning of the document. For example, meeting 3 minutes is placed before meeting 2 in the document. The team leader should prepare a basic agenda for the meeting and team members should rotate to be the minutes taker. Each group should have at least one meeting per week, and you may have multiple meetings if needed.

**Meeting 4 Week 4**

**Date and Time:** Friday, 20th September 2024 7:30 PM EST

**Place:** Virtual Remote Discord

**Participants:** Ananya Singh, Battal Cevik, Natasya Liew, Natthaphon Foithong, Poom Chantarapornrat

**Minutes taker:** Battal Cevik

**Timekeeper:** Battal Cevik

**Purpose:**

- Finalizing the details of the iteration 1 build
- Documentations to finish over weekend
- Unit tests
- Updating work
- Final additions to work on for iteration 1 release

**Agenda:**

- Updating work for the past week
- What other things need to be done for the iteration 1 release
- Support to complete the compiling
- Setting up Linting and Formatting checks to solve the build error
- Will we be able to finish compiling tonight, or push back to monday?
- On integration between SB and python - chat\_history and input parameters for the course builder.

**Discussions:**

- Discussed the docker file and docker compose file =
- Discussed the PR review and how to create small PRs in order to split approval
- Backend PR is ready for Spring Boot application
- For Iteration one team decided hardcoded for chatbot response for demo
- Postman Post call is ready for local deploy
- For demo we need to run spring boot locally using docker compose
- We need UI frontend docker compose to be ready
- Battal and AJ will work on configuration part

- Battal to create folder structure for repo
- Unit tests is done for ai-service
- Database design is ready
- Team needs to switch the token to session id.
- 2 part, first one we are going over SDD documents. The second going over the actual code and live demo
- Talked about linting and formatting the codes
- Talked about best practices writing codes
- To PR work in smaller chunks (per functionality)

#### **Key Decisions:**

- Finalized the docker file and docker compose file discussion
- Finalized PR review and how to create small PRs in order to split approval
- Backend PR is ready for Spring Boot application
- Updating Jira
- Creating test cases for functionalities
- Deciding on a linting package for each language

#### **Action Items:**

- UI will be ready for running locally Natt
- Create best practices for each component Battal
- For each component please add linting tool to format the code - everyone
- Poom to update the hardcoded JSON file
- Battal to include AJ on the CI/CD support
- Poom and Tash to finish up up on the Prompt library for next iteration before supporting front-end or SB team, but definitely the documentations
- Tash to update the JIRA

### **Meeting 3**

**Date and Time:** Sunday, 15 Sept 2024 9:35 PM EST

**Place:** Virtual Remote Discord

**Participants:** Ananya Singh, Battal Cevik, Natasya Liew, Natthaphon Foithong, Poom Chantarapornrat, Yujun Liu

**Minutes taker:** Poom Chantarapornrat

**Timekeeper:** Poom Chantarapornrat

#### **Purpose:**

- Setting up for Iteration 1
- Progress update

#### **Agenda:**

- Progress update:
  - Front-end

- Python AI Service team
  - Chatbot base (completed)
  - Dt design/Implementation (delay)
  - Mongodb – deploy push to iteration 2
  - Prompt library (TODO)
  - Goals:
    - Level 1: Use mongodb/vector to retrieve course/program info
    - Level 2: Use dt to retrieve course building function
    - Level 3: Use prompt library for logic-based query
      - E.g. what 500-level courses options can I take
      - E.g. based on my courses I've taken, which courses can I take?
      - E.g. to graduate within x-semesters how many courses and which courses should I take a semester?
- SpringBoot Java team
- Unit Test and other QA detailing
- DBMS mgmt
- Check-in/Clarification (Make sure everyone is on same page)
- Iteration 1 Planning
  - What will be in it for each section
    - Front end
    - Python ai service
    - spring boot
    - CI/CD
    - Dbms mgmt
  - Reminder writers: Natt and Battal
  - Writing update
- Next steps:
  - Integration
  - Additional features
- Improvements
- Next meeting date

### **Discussions:**

- Name changed: BU Academic Navigator (BUAN)
- Progress
  - Front-end
    - Nat is finishing the authentication page using React/Redux
    - Nat is also finishing the main chat user interface
  - Java backend

- The backend team pushed out the initial SpringBoot code to our GitHub repository.
  - Already set up the boilerplate code to get started
  - Already created the initial Flask implementation code for the backend API.
- Python AI API
  - The AI team pushed out the first model prototype which can answer some basic questions about course descriptions.
  - Working on the class recommendation algorithm with the decision tree.
- Iteration 1 Planning
  - The front-end team continues to finish the initial user interface just for the MVP
  - The AI team will create a mockup JSON format for the chatbot API for the Java backend team to test
  - Try to implement the actual Python API with Flask/HuggingFace
  - Everyone is going to have a quick meeting to integrate everything from front to back this Friday.
  - Set up docker and complete Lab 2.
  - Complete iteration 1 materials: STD, SDD, etc.
- Improvements
  - Revise our general user requirements
  - Move the tasks from GitHub issues to Jira.
  - Add more prompts to the prompt library for testing
  - Revise the SPPP file
- Next meeting date – Sunday, September 22. 9:30PM

### **Deliverable material responsibility**

- Update readme.md (Poom)
- Presentation slides iteration 1(Natt/Battal)
- Ppt video + demo (Natt/Battal)
- Update SPPP doc (YiJun)
- Update SPPP Risk mgmt (Tash)
- Complete SDD (Natt/Poom)
- Complete STD (Battal)
- Fill progress report (Everyone + Tash)
- Code (Ananya - compile)

### **Key Decisions:**

- Iteration 1
  - MVP

- Front end UI
  - Course Builder
  - Vector-based RAG
  - Postgres DBMS
  - Spring Boot Integration
  - Unit Testing and QA
- Iteration 2
  - Front end UI
  - Python AI Service:
    - Add mongodb
    - Add prompt library
  - Spring Boot Java
  - Unit Testing and QA
- Iteration 3
  - Update MVP & Testing

### **Action Items:**

- Next meeting datetime (done – Friday 20th September)
- Finish the course builder feature by next week (Tash to finish)
- Put together Python AI script to Spring boot (Poom, Ananya)
- Set up the Postgresql database (AJ)
- Finish the Chatbot UI for iteration 1 demo (Natt)
- Unit test for course builder, chatbot, sb, and other implementation for iteration 1 (Battal)
- Documentations:
  - Readme.md (Poom)
  - Presentation slides iteration 1(Natt/Battal)
  - Ppt video (Natt/Battal)
  - SPPP doc (YiJun)
  - SPPP Risk mgmt (Tash)
  - SDD (Natt)
  - STD (Battal)
  - Progress report (Everyone + Tash)
  - Code (Ananya - compile)
  - Demo video (Natt/Battal)

### **Meeting 2 Week 2**

**Date and Time:** Sunday, 08 Sept 2024 9:35 PM EST

**Place:** Virtual Remote Discord

**Participants:** Ananya Singh, Battal Cevik, Natasya Liew, Natthaphon Foithong, Poom

Chantarapornrat, Yujun Liu

**Minutes taker:** Natasya Liew

**Timekeeper:** END 10:26 PM EST

**Purpose:** Finalize Iteration 0

**Agenda:**

- Wireframe
- Architecture Design Decision
- CI/CD Setup Decision
- Automation Test Framework
- Progress Tracking Tool Decision
- Github Tickets
- User Story Points
- Database for content
- Model tuning
- SDD write up
- Finalize Business logic Proposals
- Finalize Security Design Ideas
- SDD write-up draft
- Finalize Team Name
- Finalize Product Idea
- Update on Design: Architect, Wireframe
- CI/CD Setup done
- Database done
- Model Finalizing
- Finalize Risks
- Finalize Requirements
- Finalize UI/UX Flow
- Finalize Design Pattern

**Discussions:**

- Team Name: Eagles
- On Progress Tracking Tool:
  - Using Gh issues to create tickets and Using Gh Projects to track progress
- CI/CD
  - Three branches: main, staging, development
  - Use the manual approval as auditing method
  - Turning on notifications
  - Using rollback mechanism for deployment failure
- Automation test framework

- UI: using Selenium testing
  - Backend: Rest assured for API
- Presentation schedule:
  - Iteration 0: Lead by the Java team (Ananya and Yijun/AJ)
  - Iteration 1: Natt and Battal
  - Iteration 2: Tash and Poom
  - Iteration 3: Full team
- Architecture:
  - Refer to the diagram
  - Split work division:
    - Springboot and DBMS integration: Ananya & AJ
    - Python AI service + MongoDB/DT Graph: Tash & Poom
    - Front-end: React Redux with Natt
    - QA + CI/CD mgmt: Battal
  - Do work in modular groups: arrange schedule internally; regroup on Sunday, 15th Sept after class.
  - Frontend: React Redux + API call using Fetch
  - Iteration 1:
    - Form for the input collection from user
    - MongoDB for courses based on the MSSD program
    - Create rules for the dt using a graph data structure & make
      - Level 0: program(root); Level 1: courses(child); Level 2: prerequisites(child of child)
      - Use points method and if-then statements to select the nodes to store in the variable, 'class\_to\_take'
    - Python script for putting together Llama 2 with the mongodb retrieval and function to run the course\_selector dt.
    - PostgreSQL for the user information/user mgmt
    - Springboot backend setup
    - Springboot unit test
  - Iteration 2:
    - Okta authentication
    - Change the input method for user
- Final dbms decision:
  - PostgreSQL for user mgmt
  - MongoDB for the AI chatbot feature
- Wireframe:
  - Decide to use the top right buttons, if have time allow the left navigation bar for preview of other chat history.
- On content for documents:

- If free, add as much to your part or any that you can answer. Others edit onto that.
- Same rules applies for creating ticket and project progress.

### **Key Decisions:**

- **Team name**
- **Architecture**
- **techstack/framework**
- **Methodology**
- **Best practice**
- **Work division**

### **Action Items:**

- SPPP documents
- Presentation of Iteration 0 and the video
- Progress tracking
- Lab1
- Design and code of decision tree
- MongoDB set up and data pushed
- Initial python script for AI chatbot
- Springboot/Postgresql setup
- Adding tickets to Gh issues/project
- Frontend code

### **Meeting 1**

**Date and Time:** September 3, 2024 21:25 EST

**Place:** Remote Virtual Discord

**Participants:** Ananya Singh, Battal Cevik, Natasya Liew, Natthaphon Foithong, Poom Chantarapornrat, Yujun Liu

**Minutes taker:** Natasya Liew

**Timekeeper:** **END** 21:51 EST

**Purpose:** Planning and Work Division

### **Agenda:**

- Finalizing Tech Stack
- Finalizing Work Division
- Finalizing Deadline and Meeting Schedules

### **Discussions:**

- Determining project scope (What to do)



- BU Software Development Course Builder Chatbot using Python (if have time expand in next iteration)
  - Database – MongoDB
  - Model – Llama 2
- GUI using REACT
  - wireframe
- CRUD using Java
  - Cache chat history
  - Sharing chat history via email/print
- CI/CD setup
- Architecture Design

### **Key Decisions:**

- Front-End
  - Wireframe (Natt)
- Architecture Design (Natt/Ananya?)
- CI/CD Setup (Battal)
- Automation Test Framework (Battal)
- Database (Poom/Tash)
- Model (Poom/Tash)
- Github Tickets (Tash)
- SDD write-up (AJ/Ananya)
- Next Meeting: Sunday, 8th September 20:15 EST
  - Decide on group name
  - Next steps
  - Goal: finish model

### **Action Items:**

- Wireframe
- Architecture Design
- CI/CD Setup
- Github Tickets
- Database for content
- Model tuning
- SDD write up
- List of Requirements Options
- List of Risks Options
- List of Business logic Proposals
- List of Security Design Ideas
- SDD write-up draft

**Next Meeting Agenda:**

- Finalize Team Name
- Finalize Product Idea
- Update on Design: Architect, Wireframe
- CI/CD Setup done
- Database done
- Model Finalizing
- Finalize Risks
- Finalize Requirements
- Finalize UI/UX Flow
- Finalize Design Pattern