

# StudyBuddy - Group 10 - Iteration 1 Log


---

## Members:

- Sean Lee-Wah - 221344163 - [sean1010@my.yorku.ca](mailto:sean1010@my.yorku.ca)
  - Bhavya Trivedi - 219743590 - [Bhavya23@my.yorku.ca](mailto:Bhavya23@my.yorku.ca)
  - Aqsa Malik - 221100987 - [xsa@my.yorku.ca](mailto:xsa@my.yorku.ca)
  - Vaughn - 219846914 - [vc8@my.yorku.ca](mailto:vc8@my.yorku.ca)
  - Omar Fakousa - 220609426 - [omar821@my.yorku.ca](mailto:omar821@my.yorku.ca)
  - Yash Bhupta - 219743285 - [bhupta01@my.yorku.ca](mailto:bhupta01@my.yorku.ca)
- 

## External Links:

Jira: <https://le-eecs-2311-project.atlassian.net/jira/software/projects/ID/boards/1>

Meeting Summary Video:  Customer meeting's summary video.mp4

---

## Meeting Minutes:

**Date:** Jan 18, 2026 (106 min)

**Members:** Bhavya Trivedi, Vaughn Chan, Sean Lee-Wah, Aqsa Malik, Omar Medhat Fakoussa

**Summary** Finalized the proposal for "StudyBuddy," a location-based study partner platform, distinguishing it from competitors via AI matching and local features. Established technical backup plans (e.g., Discord) and agreed to submit the project topic as a PDF while initiating client outreach.

### Key Decisions

- **Core Product:** Location-based app for finding study partners by course/interest.
- **Tech Stack:** GitHub Education pack; SQL databases; Discord as backup for messaging.
- **Submission:** Final project topic to be submitted as a PDF.

### Task Assignment:

- [ ] **Email Potential Customers** (Jan 19) – Bhavya, Vaughn, Aqsa

- [ ] **Submit Project Topic** (Jan 20) – Sean, Bhavya, Vaughn
  - [ ] **Finalize Feature List** (Jan 25) – Omar, Bhavya, Vaughn, Aqsa
  - [ ] **Define Project Scope** – Vaughn, Bhavya
  - [ ] **Setup Features & Profiles** – Sean, Vaughn, Bhavya
  - [ ] **Email Potential Clients** – Aqsa
  - [ ] **Research Competitors & Clients** – Team
  - [ ] **Refine Docs & Feature Plans** – Team
- 

**Date:** Jan 26, 2026 (78 min)

**Members:** Bhavya Trivedi, Vaughn Chan, Sean Lee-Wah, Aqsa Malik

**Summary** Concentrated on ITR0 deliverables, specifically drafting the Vision Document and User Stories. Key feature discussions included implementing a star-rating system for study partners (balancing safety and anonymity) and allowing professors or clubs (like Math Club) to host events. Logistics regarding GitHub access were addressed, and the next meeting was rescheduled to accommodate conflicts.

### **Key Decisions**

- **Features:** Agreed to implement a rating system for study sessions/partners and allow professors to create study events.
- **Workflow:** Priority placed on "Iteration Zero" requirements (Vision Document & User Stories).
- **Logistics:** Next meeting rescheduled to 6:00 PM (tomorrow); "Contacts" channel created for outreach tracking.

### **Task Assignment**

- [ ] **Prepare User Stories (1 "Big" Story Each)** (Due: Jan 27) – All Team Members
  - [ ] **Complete Vision Document** (Due: Jan 28) – Bhavya (Lead), Team
  - [ ] **Email Professor (re: GitHub/Access)** – Aqsa
  - [ ] **Follow up with Bethune & Club Infinity** – Team
  - [ ] **Send Update Email for Tuesday Meeting** – Bhavya, Vaughn
- 

**Meeting Date:** Jan 27, 2026 (Total: 89 min)

**Members:** Bhavya Trivedi, Vaughn Chan, Sean Lee-Wah, Aqsa Malik, Omar Fakousa

**Summary** Focused on structuring Iteration Zero deliverables. Established a workflow for User Stories (uploading handwritten notes to a new channel) and set a target of 24 detailed stories total (6 per iteration), categorized by themes like Privacy and Event Profiling. We

selected Netlify for hosting and defined the "Event Profiling" feature, which will link specific study sessions to courses.

### Key Decisions

- **Hosting:** Netlify in consideration as a potential hosting platform.
- **User Stories:** Target of 24 detailed stories (6 per theme/member); themes include Privacy, Session Management, and Event Profiling.
- **Feature Logic:** Event Profiling will link study events directly to specific courses.
- **Formatting:** Vision Document will use a clean, non-indented format with specific success criteria.

### Task Assignment:

- [ ] **Upload User Stories to Channel** (Immediate) – Aqsa, Team
  - [ ] **Develop 6 Detailed User Stories Each** (Due: Jan 31) – All Team Members
  - [ ] **Complete Vision Document & Video** (Due: Feb 3) – Bhavya, Vaughn
  - [ ] **Finalize Hosting Platform (Netlify)** – Bhavya, Vaughn
  - [ ] **Implement Docker** (Due: Feb 17) – Team
  - [ ] **Incorporate Success Criteria** – Vaughn, Aqsa
- 

**Date:** Jan 30, 2026 (79 min)

**Members:** Bhavya Trivedi, Vaughn Chan, Sean Lee-Wah, Aqsa Malik, Omar Fakousa

**Summary** We focused heavily on the technical architecture and user experience for the app. We mapped out the Student Profile and Homepage features—specifically adding a "wishlist" for saved events—and discussed security measures for off-campus meetups. Crucially, we finalized our tech stack to ensure we can meet our internal deadline (Friday before reading week) and agreed to use JIRA for task management.

### Key Decisions

- **Tech Stack:** We selected **React** for the frontend and **Spring Boot** for the backend (choosing them over Python for better familiarity).
- **Security:** Authentication will be strictly for confirmed York University students.
- **Architecture:** We will follow the **MVC pattern** and require class diagrams before coding begins.
- **Process:** We are adopting **JIRA** for task tracking and strictly managing "feature separation" to avoid merge conflicts.

### Task Assignment

- [ ] **Set Up JIRA Board** – Team
- [ ] **Create Class Diagrams (MVC)** – Team
- [ ] **Define Student Profile Specs** (History, Future Events, Preferences) – Team

- [ ] **Establish Internal Deadlines** (Target: Friday before Reading Week) – *Team*
- [ ] **Implement "Saved Events" Feature** – *Team*
- [ ] **Design Unique ID System for Events** – *Team*

Here is the corrected and comprehensive summary for the February 5th meeting. I have updated the details to ensure the full scope of the meeting (including the specific notes on financial iterations and project initiation documents from the shorter transcript) is covered.

---

**Date:** Feb 5, 2026 (63 min)

**Members:** Bhavya Trivedi, Vaughn Chan, Aqsa Malik, Omar Fakoussa

**Summary** We held a critical planning session for ITR0, focusing on our technical architecture and project initiation requirements. We finalized the decision to use Firebase for our database (prioritizing live updates) and agreed to structure our code using the MVC pattern. We streamlined our workflow by consolidating all documents into a single Google Drive and deciding to use Lucidchart or Figma for the required class diagrams, which Vaughn will lead.

### Key Decisions

- Database: Selected Firebase (rejected MongoDB/Supabase) to support real-time features.
- Architecture: We will use the MVC pattern. A class diagram must be completed and reviewed by the whole team before the first demo.
- Tools: We are consolidating access: Google Drive for docs, Jira for tasks, and Figma/Lucidchart for diagrams.
- Scheduling: We scheduled a mandatory check-up meeting for Monday after class.

### Task Assignment

- [ ] **Complete Architecture & Class Diagram** (Lead: Vaughn, Review: Team) – *Must be done before itr 1 deadline*
- [ ] **Update Iteration Zero Planning Doc** – *Aqsa*
- [ ] **Finish User Stories & Backlog** – *Vaughn*
- [ ] **Send Invites for Figma & Jira** – *Vaughn*
- [ ] **Investigate Kanban Connectivity** – *Aqsa*
- [ ] **Review Financial Iteration & Project Initiation Docs** – *Team*
- [ ] **Schedule Monday Checkup** – *Bhavya, Omar*

---

**Date:** Feb 11, 2026 (137 min)

**Members:** Bhavya Trivedi, Vaughn Chan, Sean Lee-Wah, Aqsa Malik, Omar Fakoussa, Yash Bhupta

**Summary** We had this meeting to complete our assigned tasks for iteration 1. Our immediate goal is to have a functional Student Profile and Authentication system running by Friday. To achieve this, we established a strict Git workflow: we will work on individual feature branches, merge into a **setup** branch for integration, and only then push to **main**. We also broke down the massive Student Profile user story into smaller, assignable sub-tasks on JIRA. We also confirmed the use of Firebase for authentication and assigned specific components (Landing Page, Profile, Events) to specific members.

### Key Decisions

- **Demo Requirements:** the app must support Sign-In and a basic User Profile.
- **Git Workflow:** We implemented a "Feature Branch → Setup Branch → Main Branch" protocol. Direct pushes to **main** are forbidden to prevent conflicts.
- **Tech Stack:** We confirmed Firebase for authentication and database management, integrated with our React/Spring Boot stack.
- **Testing Policy:** Every class and feature must have a corresponding unit test file; no code is accepted without tests.

### Task Assignment

- [ ] **Setup Database & Firebase Auth** – *Aqsa*
  - [ ] **Develop User Profile Page & Preferences** – *Omar, Sean*
  - [ ] **Develop Events Functionality & Homepage** – *Bhavya*
  - [ ] **Integrate Map APIs** – *Sean, Yash*
  - [ ] **Push Initial Setup to Main Branch** – *Vaughn*
  - [ ] **Update JIRA Board** (Break down "Student Profile" story) – *Team*
  - [ ] **Create Unit Test Files** (For all new code) – *Team*
-

# Rationale of Changes on Plan and Big Design Decisions

During Iteration 1, several adjustments were made to our original plan based on feasibility, team familiarity, and project scope constraints.

## 1. Tech Stack Change (From Python to React + Spring Boot)

Initially, multiple backend options were considered, including Python. After further discussion, the team decided to use React (frontend) and Spring Boot (backend).

### Rationale:

- Stronger team familiarity with Java from coursework.
- Better alignment with MVC architecture and structured backend development.
- Improved long-term scalability and maintainability.
- Clear separation of frontend and backend responsibilities.

Overall this decision reduced development risk and improved efficiency.

---

## 2. Hosting Decision (From Netlify to DigitalOcean)

Originally, we selected Netlify for frontend hosting due to its simplicity and built-in CI/CD integration with GitHub. However, as development progressed and backend integration became more complex, we transitioned to using DigitalOcean for hosting.

### Rationale:

- **Full-Stack Hosting Flexibility:**  
Netlify is optimized primarily for static frontend deployments. Since our application uses a React frontend and a Spring Boot backend, we required a hosting solution capable of supporting both client and server environments.
- **Backend Deployment Support:**  
DigitalOcean allows us to deploy and manage our Spring Boot backend directly on a virtual machine (Droplet), providing more control over server configuration and runtime environments.
- **Scalability and Control:**  
DigitalOcean provides infrastructure-level control (server configuration, networking, environment variables, firewall rules), which better aligns with our

long-term scalability goals.

This switch improved architectural consistency and ensured that both frontend and backend components could be hosted within a unified environment.

---

### **3. Authentication Restriction (York-Only)**

We decided authentication would be limited to confirmed York University students.

#### **Rationale:**

- Increased safety and trust within the platform.
- Reduced risk of fake accounts.
- Simplified user validation logic.
- Strengthened the value proposition as a campus-focused app.

After talking to our client, Wmmsi, the issue of safety was considered to be of high importance, so we decided to introduce authentication features as a requirement during sign up/ sign in. This aligns with our big user story of security and user trust.

---

### **4. Feature Scope Refinement**

Originally, the feature list was broader (AI matching emphasis, advanced interaction systems). During Iteration 1, we narrowed the scope to:

- User Profiles
- Course Management
- Session Logs
- Event Hosting & Cancellation
- Basic Authentication

#### **Rationale:**

- Needed to prioritize core functionality first.
- Iteration 1 focused on foundational infrastructure.
- Prevented scope creep.
- Ensured deliverables were achievable before reading week.

Advanced features (AI matching, enhanced rating systems, etc.) are reserved for future iterations.

---

## 5. Process Decision (JIRA + Strict Feature Separation)

We adopted JIRA and strict feature separation in Git, by branching for each feature.

### Rationale:

- Prevent merge conflicts.
- Improve task tracking accountability.
- Maintain structured development workflow.
- Align with professional software engineering practices.

---

## Concern with the Project or Group Members

We had a bit of trouble finding times where everybody could meet as everyone had busy schedules.

But, no other concerns with the project or any group members.

---



## Task Assignments and Estimated/Actual Time per Development:

User Story	Assignee(s)	Estimated Time	Actual Time	Status	Notes
1.2. Create User Profile Page	Omar	2 days	3 days	Completed	User profile page that user can edit
1.3. View Study Session Log	Sean	2 days	3 days	Completed	Log that shows sessions attended in the past
1.4. Manage "My Courses" List	Omar	2 days	1 day	Completed	Add and remove courses from profile
1.8. Customize Public Profile	Omar, Sean	3 days	2 days	Completed	Edit user profile (e.g. bio, study preferences)
2.1. Create and Host Events	Bhavya	4 days	3 days	Completed	Create/host events for others to join
2.3. Cancel an Upcoming Event by Hosting Party ONLY	Vaughn	1 day	1 day	Completed	Cancel a session that you have created
4.1. Create Basic Sign-Up UI Form	Aqsa	3 days	3 days	Completed	Sign-up page for user to register their information
4.2. Create Basic Sign-In UI Form	Aqsa	3 days	2 days	Completed	Sign-in page for user to enter their information from sign-up
4.3 Setup Auth Infrastructure	Yash	3 days	4 days	In progress	Authentication by email for user to log-in

<input checked="" type="checkbox"/> ID-42	1.2. Create User Profile Page	DONE ▾	-	=	OF
<input checked="" type="checkbox"/> ID-44	1.3. View Study Session Log	DONE ▾	-	=	SL
<input checked="" type="checkbox"/> ID-45	1.4. Manage "My Courses" List	DONE ▾	-	=	OF
<input checked="" type="checkbox"/> ID-54	1.8. Customize Public Profile	DONE ▾	-	=	SL
<input checked="" type="checkbox"/> ID-36	2.1. Create and Host Events	DONE ▾	-	=	BT
<input checked="" type="checkbox"/> ID-38	2.3. Cancel an Upcoming Event by Hosting Party ONLY	DONE ▾	-	=	VC
<input checked="" type="checkbox"/> ID-35	4.1. Create Basic Sign-Up UI Form	DONE ▾	-	=	AM
<input checked="" type="checkbox"/> ID-55	4.2. Create Basic Sign-In UI Form	DONE ▾	-	=	AM
<input type="checkbox"/> <input checked="" type="checkbox"/> ID-56	4.3 Setup Auth Infrastructure <a href="#">🔗</a>	IN PROGRESS ▾	-	=	YB ...

Spaces
W26\_2311Z\_GROUP\_10\_StudyBuddy
Summary Pages Backlog Board Code Timeline Forms +

Q Search board
BT AM OF SL VC YB Filter

BIG USER STORIES 4

1. Student Profile  
☒ ID-24

2. Events  
☒ ID-31

3. Communication  
☒ ID-29

4. Application Login and User Security  
☒ ID-30

+ Create

UP NEXT 19

1.5. Send Connection Request  
☒ ID-47

1.6. Edit Study Preferences  
☒ ID-50

1.7. View Network List  
☒ ID-52

2.2. Join Existing and Upcoming Events  
☒ ID-37

2.4. Privacy Policy - Events Specific: All attendees profiles must be visible  
☒ ID-41

2.5. Cancel Joining an Event by the attendee user themselves ONLY  
☒ ID-49

2.6. Events Review Opens once events finishes  
☒ ID-58

2.7. Event Reviews Commenting is only available to attending attendees  
☒ ID-59

2.8. Events Reviews to have follow up reply feature to allow all users interaction.  
☒ ID-60

3.1. View possible sessions on map

IN PROGRESS 1

4.3 Setup Auth Infrastructure  
☒ ID-56

REVIEW

DONE 8

1.2. Create User Profile Page  
☒ ID-42

1.3. View Study Session Log  
☒ ID-44

1.4. Manage "My Courses" List  
☒ ID-45

1.8. Customize Public Profile  
☒ ID-54

2.1. Create and Host Events  
☒ ID-36

2.3. Cancel an Upcoming Event by Hosting Party ONLY  
☒ ID-38

4.1. Create Basic Sign-Up UI Form  
☒ ID-35

4.2. Create Basic Sign-In UI Form  
☒ ID-55

## Development Tasks per User Story (Estimated/Actual Time):

### 1.2. Create User Profile Page

Task	Estimated Time	Actual Time
Edit bio and study preferences	2 hours	3 hours
Display "My Courses" summary	1 hour	2 hours
Display session log summary	1 hour	1 hour
Style the profile page	1 hour	1 hour

### 1.3. View Study Session Log

Task	Estimated Time	Actual Time
Add course filter dropdown	2 hours	1 hour
Display total session time	1 hour	2 hours
Frontend component on profile page	1 hour	1 hour

### 1.4. Manage "My Courses" List

Task	Estimated Time	Actual Time
Backend for adding courses	2 hours	1 hour
Backend for deleting courses	2 hours	2 hours
Frontend component on profile page	2 hours	3 hours

## 1.8. Customize Public Profile

Task	Estimated Time	Actual Time
Update study preferences	2 hours	3 hours
Edit bio	2 hours	2 hours
Frontend and backend polish	1 hour	2 hours

## 2.1. Create and Host Events

Task	Estimated Time	Actual Time
Create event frontend	2 hours	3 hours
Create event backend	2 hours	4 hours
Implement API to store events	3 hours	3 hours

## 2.3. Cancel an Upcoming Event by Hosting Party ONLY

Task	Estimated Time	Actual Time
Frontend for “delete” button to remove event	3 hours	3 hours
Backend to remove event from database	4 hours	5 hours
Update session log	3 hours	2 hours

## 4.1. Create Basic Sign-Up UI Form

Task	Estimated Time	Actual Time
Frontend for sign-up page to allow user to enter information	3 hours	3 hours
Backend for processing information and storing it in database	4 hours	5 hours
Redirect user upon successful sign up	3 hours	2 hours

## 4.2. Create Basic Sign-In UI Form

Task	Estimated Time	Actual Time
Frontend for sign-in page	3 hours	3 hours
Backend to confirm user details match the one found in database	3 hours	3 hours
Redirect to event page upon success	1 hour	2 hours

## 4.3 Setup Auth Infrastructure

Task	Estimated Time	Actual Time
Setup Firebase database	2 hours	3 hours
Add email verification	3 hours	2 hours
Test sign-in/sign-up	1 hour	2 hours
Backend and frontend components	5 hours	6 hours

---

## Next Steps:

1. Decide and assign tasks for iteration 2
2. Standardize the tech stack used across the detailed user story work items.
3. Implement the authentication functionality to our working MVP frontend as that's a task carried over from ITR1 development.
4. Create additional unit tests to ensure proper functionality and code coverage.