

## New Scope for Project - ITR3 & Deliverable 2

This document outlines the revised scope of our project following the team meeting with **Prof. Hadi**. The new scope reflects the work that will be completed in **Iteration 3 (ITR3)** and **Deliverable 2**. The objective is to ensure that each team member is accountable for their assigned user stories and contributes toward a fully functional and well-documented final product.

### Updated User Story Assignments

User Story	Assigned To	Status (As of March 18th)	Expected Completion
Course Calendar	Abdulkadir Ahmed	Partially Complete (ITR2 Ready)	ITR3 Enhancements (March 31)
User Accounts (new)	Abdulkadir Ahmed	Not Started	March 25 (Barebones Login/Signup, connected to Render SQL DB)
Group Chat (Without user accounts)	Simmonly & Digvijay	Incomplete, missing frontend functionality	March 20-21
Group Chat (With user accounts)	Simmonly & Digvijay	Not Started	March 28
AI Advising Chatbot	Fardad Rashidian	Ongoing (Needs Completion for ITR3)	ITR3 Enhancements (March 31)

Details of Each User Story for ITR3 (As of March 18th) (Details are low but please read ITR3 and previous iterations requirements ITR1 ITR2)

#### 1. Course Calendar (Abdulkadir Ahmed)

Current Status:

- Implemented course entry system, allowing users to input their course name, professor, time, and day.
- Courses persist and are displayed dynamically in a scrolling UI.
- Users can plot their courses into a calendar view (24-hour format).

Enhancements for ITR3:

- UI/UX Improvements (Better spacing, possibly limiting hours displayed).
- Potential Backend Enhancements (If time permits, may improve database interactions).
- Bug Fixes & Testing.

This feature is the most complete, close to ITR3-ready.

Expected Completion: March 31

## 2. User Accounts (Abdulkadir Ahmed)

Scope:

- Implement barebones Signup/Login (No authentication, just storing user details in a PostgreSQL database on Render).
- This will allow users to log in and access personalized features (for future use i.e group chat which requires it).

Expected Completion: March 25

## 3. Group Chat (Simmonly & Digvijay)

Current Status:

- No frontend implementation. Backend has some progress.

No user account group chat feature which is barebone should be completed by March 20-21.

ITR3 Requirements:

- Fully working frontend UI for group chat to allow users to send and receive messages.
- Basic text-based messaging functionality (at minimum)
- Connect to the Render backend if possible.

Progress checks required to ensure completion by March 31 (Checks on March 20-21 for version without User Accounts and March 28 with User Accounts)

Expected Completion: March 20-21 and Expected Completion: March 28

## 4. AI Advising Chatbot (Fardad Rashidian)

Current Status:

- Backend implemented (accepts user PDFs and summarizes them using GPT-4).
- Allows user queries about their PDF content.

ITR3 Completion Goals:

- Final testing and bug fixes.
- Ensure the chatbot is fully integrated into the frontend UI.
- User experience improvements (if time permits).

Should be fully working by March 31.

## Progress Check Intervals

March 22: First check-in on team progress.

March 25: User Accounts feature completed (Abdulkadir).

March 28: Final pre-delivery check (Make sure all major features are functional).

March 31: ITR3 & Deliverable 2 Submission & Presentation.

---

IMPORTANT:

1. All team members are expected to complete their assigned tasks by their respective deadlines.
2. Team progress will be reported to Prof. Hadi and T.A every 3 days.
3. Final Submission & Presentation on March 31.

## Merge Requests Policy – ITR3 New Scope

To ensure smooth collaboration and efficient code review, the following merge request (MR) guidelines must be followed moving forward for ITR3 and Deliverable 2:

---

## Merge Request Guidelines

### 1. One User Story, One Branch

- Each developer must work on a single feature per branch.
- Do not create multiple unnecessary branches for minor changes.

### 2. Merge Request Only When Feature is Complete

- Only submit a merge request when your feature is fully implemented and ready for testing.
- No half-completed merge requests—test locally first before submitting.

### 3. Request a Review Before Merging

- Every merge request must be reviewed before merging.
- Tag @AbdulkadirAhmed when the branch is complete and ready for testing.
- Provide clear descriptions in the merge request (what was added, what was fixed).

### 4. No Duplicate Work

- Do not duplicate efforts by working on features that already exist or are in progress.
- Check GitHub before starting a new branch.

### 5. Keep Branches Organized

- Use descriptive names for branches (e.g., `feature-group-chat-ui`, `fix-advising-backend`).
- Delete old/unused branches once merged to keep the repo clean.