SLIIT
Discover Your Future

BSc (Hons) in Information Technology
Year 3
Project Requirements

IT3040 – ITPM **Semester 1**

1. Project Overview

Each group must propose a **practically feasible** project that delivers clear value to **university students**. The proposed solution may take the form of a **web, mobile, or desktop application**, and should address a real need relevant to student life. Examples include (but are not limited to):

- An education-focused platform for sharing learning resources, delivering lessons, and supporting online classes
- A student support application that helps peers collaborate and assist one another with day-to-day academic or personal activities
- A multiplayer online game designed for friends to play together remotely
- A social matching application that helps students connect with others who share similar interests or goals

The project scope must be appropriate for a team of **up to four members** and must be achievable within an **11-week** timeframe.

---

2. Assessment Structure and Milestones

The project will be evaluated through **five main assessments** conducted throughout the semester. These assessments are designed to evaluate both **group progress** and **individual contribution** at key stages of the project lifecycle.

---

3. Individual Contribution Expectations

The project must be undertaken by a group of **up to four students**. However, **individual contributions will be evaluated** during progress reviews and the final presentation. To ensure equitable contribution, each student must:

- Take ownership of **at least one independently demonstrable feature/module**, and
- Ensure the component includes appropriate **UI design**, **database integration (where relevant)**, **input validation**, and **testing**.

Note: Group members are **not required** to implement identical features (e.g., CRUD operations, report generation, etc.). Instead, each member's component must demonstrate **meaningful functionality**, **technical effort**, and **integration** with the overall system.

---

4. Tools and Technology Selection

There are **no marks allocated specifically** for the tools, frameworks, languages, or platforms selected. Therefore, each team may decide on:

- The application type (web/mobile/desktop)
- The technologies and tools used
- The overall implementation approach

However, the selected technologies must support successful delivery within the given time frame and project scope.

5. Proposal Requirement: Responsibility Matrix

At the **Proposal stage,** each group must submit a **Responsibility Matrix** identifying each member's ownership and scope of work.

The Responsibility Matrix must include the following details for each student (use the template provided in **Appendix 1**):

- Student Name (as per SLIIT ID)
- Registration Number
- Name of the responsible component(s)/feature(s)
- A brief description of the component(s)/feature(s), clearly stating:
    - **Inputs** (what the user/system provides)
    - **System outputs** (what the feature produces)
    - **Intended user outcome** (what the feature enables for users)

---

6. Progress 1 Requirement: UI Business Rules Document

At **Progress Review 1**, each group must submit a document clearly stating the **business rules** (**refer to Appendix 2**) related to each student's assigned UI(s) and component(s).

Accordingly, using the template provided in **Appendix 3**, each student must document:

- All UI screens/pages/forms under their responsibility
- Business rules applicable to each UI

---

7. Student Readiness for Assessments

It is the responsibility of each student to clearly communicate and demonstrate the tasks they have completed during each assessment within the allocated time. Students are expected to be well prepared to:

- Explain their responsible component(s)
- Demonstrate working functionality
- Justify design and implementation decisions
- Answer questions related to their own work

8. Final Presentation Evidence Requirements

At the **final presentation**, each student must demonstrate evidence of their individual contribution, including the use of:

- **Version control (Git)** showing meaningful commits related to their module(s)
- A **project management tool** (e.g., Trello/Jira/GitHub Projects) demonstrating planning and task ownership
- An **automated testing tool** (e.g., Playwright/Cypress for web, Appium/Detox for mobile, WinAppDriver/FlaUI for desktop) used to validate key features and user journeys of the proposed solution.

---

9. Bonus Mark Allocation

A bonus mark allocation mechanism will be implemented to recognize exceptional performance. Following an initial shortlisting based on the **total marks** obtained **up to the final presentation**, **30 projects** will be identified as the best-performing projects. From these, a **maximum of 10 projects** will be selected based on a separate assessment.

Each selected group will be **awarded 30 bonus marks (out of 100)**, which will be credited to the **Final Component** of the module. With the allocation of these bonus marks, the maximum mark a student can obtain for the Final Component is 100.

Appendix 1: Responsibility Matrix Template for the Proposal Presentation

Student name	Registration number	Responsible component(s) name	Description of responsible component(s) (Inputs, system outputs, intended user outcome)
A. de Villiers	IT28123411	Peer tutoring: Session Booking Module	**Inputs:** ▪ Student selects a subject, preferred time slot, and tutor ▪ Student submits the booking request **Outputs:** ▪ Booking is saved successfully in the database ▪ Confirmation message is displayed to the student ▪ Tutor receives an in-app notification (or email notification) **Intended user outcomes:** ▪ Students are able to schedule tutoring sessions efficiently ▪ Reduced coordination time and fewer missed tutoring sessions
V. Kohli	IT28123421	Game module: Lobby creation & invite system	**Inputs:** ▪ Player selects the preferred game mode and lobby settings (e.g., public/private, number of players) ▪ Player clicks “Create Lobby” to initiate a new session **Outputs:** ▪ A unique Lobby ID (or Room Code) is generated by the system ▪ The lobby is created and stored in the database with the selected settings ▪ An invite link or invite code is displayed for sharing with friends **Intended user outcomes:** ▪ Players can create and share multiplayer lobbies quickly and easily ▪ Friends can join the same game session reliably using an invite code/link ▪ Players can start a match without manual coordination or repeated setup steps

Appendix 2: Business Rules

Business rules refer to the functional conditions that control how each UI operates and how the system should respond to user actions. For each UI under a group member's responsibility, the business rules must include relevant access restrictions (who can use it), validations (mandatory fields and allowed values), workflow rules (status changes and action limits), and system behavior (what happens when the user submits valid or invalid inputs). Business rules must be specific and testable.

A good business rule answers questions like:

- **Who** can do this action?
- **When** can they do it?
- **What** conditions must be true?
- **What** should happen if rules are violated?
- **What** should the system save/update/show?

Business rules should include most of the following where applicable:

A) Access Control Rules

- Who can view this screen?
- Who can perform actions?
- Role-based restrictions (Student/Tutor/Admin/Host)

Example: Only the host can start the game.

B) Validation Rules (Input Constraints)

- Mandatory fields
- Allowed values/ranges
- Format rules (email, phone, time, password strength)

Example: Date must be in the future.

C) Process/Workflow Rules

- The sequence of actions
- Status transitions (Pending → Approved → Completed)

Example: Booking cannot be edited once confirmed.

D) Capacity / Limits / Availability Rules

- Maximum allowed items
- Slot conflicts
- System limits

Example: Lobby cannot exceed 8 players.

E) Data Consistency Rules

- No duplicates
- Unique IDs/codes
- Prevent conflicting records

Example: A student cannot book the same session twice.

F) Notification or System Response Rules

- Confirmation messages
- Error handling behavior
- Notifications triggered

Example: When a lobby is created, an invite code must be shown.

G) Security & Integrity Rules (optional)

- Session expiry
- Prevent unauthorized access
- Prevent manipulation

Example: Invite code must expire after the lobby starts.

Appendix 3: UI and Business Rules Template for the Progress 1 Presentation

Student name	Registration number	Responsible component(s) name	UIs related to responsible component(s)	Business rules applicable to each UI
A. de Villiers	IT28123411	Peer tutoring: Session Booking Module	**UI 1:** Session booking form 	▪ Only logged-in students can create bookings. ▪ Required fields: subject, tutor, date, start time, duration. ▪ Selected date/time must be in the future. ▪ Duration must be within allowed range (e.g., 30–180 minutes). ▪ Time slot must not conflict with tutor availability or existing bookings. ▪ Display clear validation messages for invalid inputs.
			UI 2: Booking confirmation/ summary screen 	▪ Show booking ID, tutor name, subject, date/time, status. ▪ Status must reflect system state (Pending/Confirmed/Cancelled). ▪ Students can cancel only before a cutoff time (e.g., 2 hours before session).
			UI 3: Student booking history screen 	▪ Students can view only their own bookings. ▪ Past sessions are read-only and cannot be modified. ▪ Filters available (Upcoming/Past/Cancelled).
			UI 4: Tutor booking requests screen 	▪ Tutor can view only bookings assigned to them. ▪ Tutor can accept/reject requests (if approval flow exists). ▪ Rejection must capture a reason (optional). ▪ System updates status and notifies student.
V. Kohli	IT28123421	Game module: Lobby creation & invite system	**UI 1:** Create lobby screen 	▪ Only authenticated players can create a lobby. ▪ Required inputs: game mode, lobby type (public/private), max players. ▪ Max players must be within allowed limits (e.g., 2–10). ▪ For Private lobbies, system must generate a unique invite code. ▪ Lobby must be created with status “Waiting for players”.
			UI 2: Lobby waiting room (Host view) 	▪ Host can see list of joined players and lobby settings. ▪ Host can start the game only when minimum players joined. ▪ Host can remove players (optional rule). ▪ Host can cancel the lobby; cancellation removes lobby and notifies players.
			UI 3: Join lobby screen 	▪ Player must enter a valid lobby code (for private lobbies). ▪ System must reject invalid/expired codes. ▪ Joining is blocked if lobby is full. ▪ System must prevent the same player joining twice.
			UI 4: Invite link/code display screen 	▪ Invite code/link must be visible only to host (or lobby members). ▪ Invite code should be copyable/shareable. ▪ Code must remain valid only while lobby status is “Waiting”.