|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk ID | Description | Likelihood | Impact | Mitigation Strategy |
| R1 | Group member drops out or becomes unresponsive | Medium | High | Define backup roles early; redistribute workload dynamically |
| R2 | Communication breakdown (e.g., unclear tasks or misalignment) | Medium | Medium | Use Agile tools like Trello/Taiga, have weekly sync calls |
| R3 | Git merge conflicts or overwrites | Low | High | Strict branch-per-feature workflow, use pull requests only |
| R4 | Missed deadlines on milestones | Medium | High | Set internal buffer deadlines; weekly status meetings |
| R5 | Technical/tooling issues (e.g., GitHub downtime, MS Project crashes) | Low | Medium | Keep local backups, export task plans regularly |
| R6 | Scope creep due to enthusiasm or user feedback | Medium | Medium | Lock features by Week 4; require team vote for new features |
| R7 | Team workload spikes from overlapping uni assignments | High | Medium | Plan around assessment calendar; allow flexible task assignments |
| R8 | Incomplete or insufficient testing | Medium | Medium | Schedule test sessions; assign dedicated test ownership |
| R9 | Front-end/back-end integration bugs | Medium | Medium | Define API contracts early; conduct integration testing weekly |
| R10 | Misestimation of development effort | Medium | High | Use function point estimation; re-evaluate task loads regularly |

RISK MANAGEMENT PLAN

## Project Scope Statement

****Project Title-** FlightEZ – Customer & Flight Management System**

**Project Objective**

To design, develop, and test a full-stack software system that enables customers to register, search and book flights, manage seat reservations, and purchase in-flight services (e.g., food, drinks), while also providing administrative tools for managing flights and bookings.

**Scope Description**

This project will deliver a locally hosted, web-based platform that supports both customer-facing and administrator-facing interfaces. Core components include account management, flight booking, reservation control, and service purchases. The solution will follow a modular architecture with back-end, front-end, and database layers, and will be developed using version-controlled collaborative tools.

**In Scope**

- User registration, login, and profile management

- Flight search with filters (origin, destination, date)

- Flight booking and real-time seat selection

- Interface for purchasing in-flight services

- Admin dashboard to manage:

- Flight schedules

- Customer records

- Booking data

- Fully responsive front-end (desktop only)

- Back-end API with mock flight data (no real integrations)

- Local test environment setup

- Documentation:

- Technical documentation

- User manual

- Testing (unit + system-level)

Out Of Scope

- Payment processing (e.g., PayPal, Stripe)

- Integration with real airline APIs

- Mobile application development

- Cloud hosting or public server deployment

- Ongoing post-submission support or updates

Deliverables

- A functional web application running on a local test environment

- Technical documentation and user guide

- GitHub project repository with commit logs

- Milestone tracking (Microsoft Project or Agile tool)

- Final group presentation/demo

Constraints

- Team of 4 student developers

- Must use

- **GitHub** **for version control**

- **Microsoft Project** **for schedule tracking**

- **Taiga/JIRA** **optionally for Agile boards**

- All development must run on local-host/test machine

- Final scope must be frozen by **Week 4**

Assumptions

- All team members are available for weekly meetings and work

- Mock flight data will be generated or preloaded manually

- All assets (UI, icons, content) are created by the team

- Tools and frameworks used are free/open-source or provided by the university