# 6. Development Plan & Timeline

This section defines how the team will execute Deliverable 2 (Software Design Specification) as a continuation of the existing SRS. It covers task partitioning, responsibilities, milestones, and the submission workflow.

## 6.1 Roles & Responsibilities

**Team members:**

• Matthew Gallenberger – Person A (Architecture & Diagrams)

• Ryan Schindler – Person B (Class & Architecture Descriptions)

• Nikola Milosavljevic – Person C (Plan, Timeline, Repo Setup)

**Partitioning of tasks:**

* Person A – Architecture & UML Diagrams
* Create Software Architecture Diagram (components & connectors: Web UI, App Server, DB, Payment Provider, Notification Service).
* Create UML Class Diagram (Screening, Auditorium, Seat, Order, Ticket, Payment, User Roles, Policy Values).
* Export diagrams as PNG/SVG and deliver assets for integration.
* Person B – Class & Architecture Descriptions
* Write detailed class descriptions (attributes with datatypes; operations with signatures/parameters).
* Write Architecture explanation (component responsibilities and interactions).
* Own editorial pass & final merge into master SRS (DOCX/PDF).
* Person C – Development Plan & Timeline (this section) + Repo Setup
* Initialize GitHub repository and upload prior SRS document.
* Write this Development Plan & Timeline section and keep it updated based on team inputs.
* Coordinate submission checklist and ensure each member has ≥1 commit.

## 6.2 Design Deliverables & Acceptance Criteria

• Architecture Diagram: shows all major components and connectors; exported as image; references match class/section names.

• UML Class Diagram: includes class names, attributes, operations, and relationships (associations/aggregations/inheritance).

• Class & Architecture Descriptions: for every class, list attributes (with datatypes) and operations (with signatures) and describe purpose; provide 1–2 paragraphs that explain the overall architecture.

• Development Plan & Timeline: this section; concise and actionable.

• Final SRS: single DOCX and exported PDF in the repo under /deliverables, with images embedded and section numbering consistent.

## 6.3 Timeline & Milestones

Target schedule (America/Los\_Angeles). Weeks are guidelines; members may work asynchronously as long as handoff dates are met.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Week / Dates | Owner | Focus | Milestone | Handoff Output |
| Week 1 (Sep 29 – Oct 5, 2025) | Matthew (A) | Architecture Diagram + UML Class Diagram | Draft diagrams ready | PNG/SVG exports + source files |
| Week 2 (Oct 6 – Oct 12, 2025) | Ryan (B) | Class & Architecture Descriptions | Complete draft text | DOC/MD with class tables & arch description |
| Week 3 (Oct 13 – Oct 19, 2025) | Ryan (B) + All | Editorial & Merge; light review by all | Final SRS (DOCX/PDF) assembled | Merged DOCX + exported PDF in repo |

## 6.4 Workflow & Submission

1) Repository: GitHub repo with /src (assets), /design (diagrams), /docs (SRS working files), /deliverables (final DOCX/PDF). https://github.com/KypleeFearz/CS250\_Group7

2) Branching: simple trunk-based; optional feature branches per person (e.g., feature/uml-diagrams, feature/class-descriptions).

3) Commits: each member must have ≥1 commit (add diagrams, text, or minor fixes).

4) Integration: Person B merges diagrams + text into the SRS template and exports PDF.

5) Submission: push final DOCX/PDF to /deliverables and submit the GitHub link in the course system.

## 6.5 Risks & Mitigations

• Late handoff of diagrams → Mitigation: placeholders for class names so B can proceed; finalize images before Week 3.

• Format inconsistencies → Mitigation: use the same SRS template; Person B owns one editorial pass in Week 3.

• Missed commit credit → Mitigation: add a small personal change (typo fix or minor edit) if needed before submission.