**Georgi Tinchev**

**Individual Project Plan**

***“Stream Sage”***



**Gitlab Repo Link:**[**https://git.fhict.nl/I524441/individual-project-sem2**](https://git.fhict.nl/I524441/individual-project-sem2)

**Table of Contents:**

1. **Introduction**
   * 1.1 Project Contacts
   * 1.2 Current Situation
   * 1.3 Problem Description
2. **Project Details**
   * 1.4 Project Goal
   * 1.5 Deliverables
   * 1.6 Non-Deliverable
   * 1.7 Constraints
   * 1.8 Risk Analysis
3. **Phasing & Timeline**
   * 1.9 Phasing
   * 1.9 Timeline
4. **Gantt Chart**
   * 6.1 Phasing Gantt Chart

**1.1 Project Contacts**

* Project Owner: Georgi Tinchev.
* Client: Bhardwaj, Sachin S., E. Mladenovska, R. Avetyan.
* Email: [georgi.tinchev.124@gmail.com](mailto:georgi.tinchev.124@gmail.com)
* Phone: +359892984923
  1. **Current Situation**
* The project aims to develop a streaming service app with innovative features beyond traditional movie platforms.
  1. **Problem Description**
* Lack of immersive and educational experiences in conventional movie streaming platforms.
  1. **Project Goal**
* Redefine movie streaming by seamlessly blending entertainment with education, social engagement, and critiques.

\* 1.2 through 1.4 are too vague and just one sentence

**1.5** **Deliverables**

* Customer-facing movie streaming web app.
* Administrative control facing desktop app.
* SQL database for relational data storage.
* Login functionality.
* Registration functionality.
* Search functionality.
* Movie recommendation algorithm.
* Leaderboard recommendations algorithm.
* User Reviews and Ratings.
* UML Diagram
* Project Plan & URS

\* Functionalities not mixed with deliverables

**1.6** **Non-Deliverable**

* Complicated UI navigation.
* Annoying non-consensual autoplay.
* Advanced social media-like functionality.

**1.7 Constraints**

* Budget: N/A
* Timeline: 16 Weeks
* Due date: Jun 14th.
* Programming Language Constraints – C# etc.
  + Constraints are known not etc.

**1.8 Risk Analysis**

Potential Risks:

1. Change in technologies affecting app development.
2. Time constraints affecting completion of optional features.
3. Bugs arising during development leading to delays.
   * Also how to mitigate the risks not just the risks

**1.9 Phasing & Timeline**

1. Ideation and Planning (Weeks 1-3)

* Develop Ideation document through discussions.
* Submit Ideation document by Friday, Week 1.

2. Requirements and Design (Weeks 4-10)

* OOD:
  + Work on UML class diagram & desktop application.
  + Seek OOD teacher's feedback.
  + Submit intermediate version by Friday, Week 5.
  + Attend formal feedback sessions.
* WAD:
  + Start web application development.
  + Seek WAD teacher's feedback.
  + Submit intermediate version by Friday, Week 6.
  + Attend formal feedback sessions.
* WKS:
  + Begin Project plan.
  + Seek WKS teacher's feedback.
  + Submit Project plan by Friday, Week 3.

3. Implementation and Testing (Weeks 11-15)

* OOD:
  + Continue UML & desktop development.
  + Seek OOD teacher's feedback.
  + Submit intermediate version by Friday, Week 10.
  + Attend formal feedback sessions.
* WAD:
  + Continue web application development.
  + Seek WAD teacher's feedback.
  + Submit intermediate version by Friday, Week 11.
  + Attend formal feedback sessions.
* WKS:
  + Continue URS development.
  + Seek WKS teacher's feedback.
  + Submit URS by Friday, Week 7.
  + Start Test plan & database diagram.
  + Seek WKS teacher's feedback.
  + Submit Test plan & database diagram by Friday, Week 12.

4. Finalization and Submission (Week 16)

* Finalize Project plan, URS, Test plan & database diagram.
* Request feedback.
* Submit final work by Friday, Week 16.
  + The current phasing is too similar to canvas details, make it more project oriented and don’t get it mixed up.

A graph with text and numbers

Description automatically generated with medium confidence**2.1 Phasing Gantt Chart**

Z

* + Repetition of headers makes it unclear what you are working on when, URS Dev repeats twice at different days for example.