**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 current landscape of movie streaming platforms lacks depth in providing users with immersive and educational experiences. Traditional platforms primarily focus on entertainment, leaving a gap in fulfilling the educational and social aspects of content consumption.
  1. **Problem Description**
* The prevalent issue lies in the one-dimensional nature of existing movie streaming platforms. Users seek more than just entertainment; they desire enriching experiences that combine learning, social interaction, and critical engagement.
  1. **Project Goal**
* The project aims to redefine the concept of movie streaming by integrating innovative features that transcend the boundaries of traditional platforms. By seamlessly blending entertainment with educational content, social engagement, and critical analysis, the goal is to create a holistic streaming experience that caters to diverse user preferences and interests.

**1.5** **Deliverables**

* Customer-facing movie streaming web app.
* Administrative control facing desktop app.
* SQL database for relational data storage.
* UML Diagram.
* Project Plan & URS.

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

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

**1.7 Constraints**

* Use FHICT GitLab for version control.
* Web app in ASP.NET Core Razor Pages, desktop app in .NET Core Windows Forms.
* Avoid Entity Framework or similar ORMs.
* Database in MSSQL with many-to-many relationships.
* Implement non-trivial functionality using C#.

**1.8 Risk Analysis**

Potential Risks:

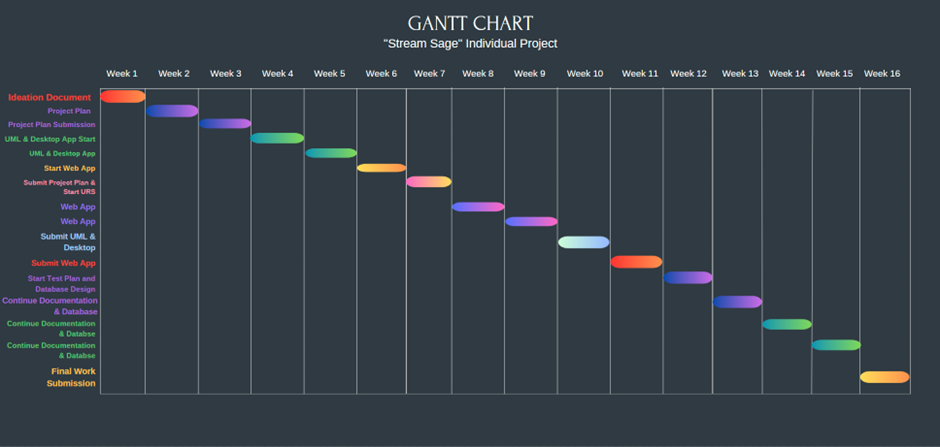
* Change in technologies affecting app development.
* Time constraints affecting completion of optional features.
* Bugs arising during development leading to delays.

Mitigation Strategies:

* Regularly monitor technological advancements.
* Prioritize essential features to ensure timely completion.
* Implement thorough testing procedures to detect and resolve bugs promptly.

**1.9 Phasing & Timeline**

1. Ideation and Planning (Weeks 1-3)
   * Develop project goals and requirements.
   * Submit Ideation document by end of Week 1.
   * Draft Project plan by end of Week 3.
2. Requirements and Design (Weeks 4-10)
   * Work on UML class diagram and desktop application design.
   * Submit UML class diagram and desktop application design by end of Week 5.
   * Work on Web Application.
3. Implementation and Testing (Weeks 11-15)
   * Implement desktop and web application features.
   * Conduct unit testing and peer reviews.
   * Submit intermediate versions of applications for testing.
   * Finalize Test plan & database diagram by end of Week 12.
4. Finalization and Submission (Week 16)
   * Finalize project documents based on feedback.
   * Conduct thorough testing and address issues.
   * Submit final project deliverables by end of Week 16.

**2.0 Phasing Gantt Chart**