Quiz Odyssey

Project Plan

Group: 26

Eren Taşdemir	150200035
Berkant Bakışlı	150200069
Karol Jan Charchut	912310003
Pijus Jonas Navasaitis	912310007
Egi Gjineci	150190910

Table of Contents

Introduction

Scope

The scope of the trivia game should include some key functionalities: being able to answer and compete in daily challenges, get achievements, and progress through experience levels. Also, players should be able to customize their personal accounts. The scope also includes potential use in the educational system.

Daily Challenges:

- Implement a system of daily challenges, where users can participate in a variety of trivia questions on a daily basis.
- Track and reward user participation and performance in daily challenges to maintain user engagement.
- Design and develop a dynamic question pool that provides fresh and challenging content each day.

Progression and Achievements:

- Create a progressive structure within the game, allowing users to advance through levels or ranks based on their performance.
- Incorporate a system of achievements and rewards to motivate users to continue their engagement and track their progress.

Customization and Personalization:

- Develop a user-friendly interface that enables customisation of user profiles, avatars, and preferences.
- Implement a feature to store and showcase user statistics and achievements.
- Allow users to see other players' profiles, follow them and compare their own statistics.

Potential Use in Educational Systems:

- Ensure the trivia game aligns with educational standards and objectives, making it suitable for formal and informal educational settings.

Deliverables

There are four main deliverables that we are required to ensure a functional mobile quiz experience.

Delivery number	Explanation	Date of delivery
1.	Project plan	10.11.2023
2.	Software Requirements Specification	17.11.2023

3.	Software Requirement Analysis	24.11.2023
4.	Database Design and Implementation	27.11.2023
5.	UI Design and Implementation	30.11.2023
6.	Back-end Design and implementation	07.12.2023
7.	Test Report	25.12.2024
8.	Source Codes	26.12.2023
9.	Final Demo	27.12.2023

Functional Requirements

1. Project Plan:
   - Define project objectives and scope.
   - Allocate resources, including team members and budget.
   - Create a project schedule with milestones and deadlines.
   - Develop a risk management plan.
   - Define roles and responsibilities within the project team.

2. Software Requirements Specification:
   - Specify the app's features, user interfaces, and interactions.
   - Describe any integration requirements with third-party services or APIs.

3. Software Requirement Analysis:
   - Analyze and refine the gathered requirements.
   - Prioritize requirements based on their importance and impact.
   - Resolve conflicts or ambiguities in the requirements.

4. Database Design and implementation:
   - Create an entity-relationship diagram (ERD) to model the database schema.
   - Define data tables, fields, relationships, and constraints.
   - Specify data storage and retrieval requirements.
   - Optimize the database design for efficiency and performance.

5. UI Design and implementation:
   - Create wireframes and mockups for the app's user interface.
   - Design the app's visual elements, including layouts, colour schemes, and typography.
   - Develop user-friendly navigation and user experience (UX) design.
   - Ensure responsive design for different screen sizes and devices.
   - Define user interface guidelines and design principles.

6. Back-end Design and implementation:
   - Determine the server architecture, including the choice of technologies and frameworks.

- Define the user-side logic and data processing flows.
- Specify APIs and endpoints for communication between front-end and back-end components.
- Develop data access and storage mechanisms.
- Plan for security measures, such as authentication and authorization.

7. Test Report:
   - Define test cases and test scenarios based on the software requirements.
   - Execute test cases and record test results.
   - Identify and document any defects or issues found during testing.
   - Provide an overview of test coverage and test completeness.
   - Evaluate the software against predefined acceptance criteria.

8. Source Codes:
   - Write clean and well-documented code based on the software requirements.
   - Version control using a source code management system.
   - Comment code for clarity and maintainability.

9. Final Demo:
   - Prepare a working version of the app for presentation.
   - Demonstrate the app's features, functionality, and user interface.
   - Showcase any specific features or improvements made during development.

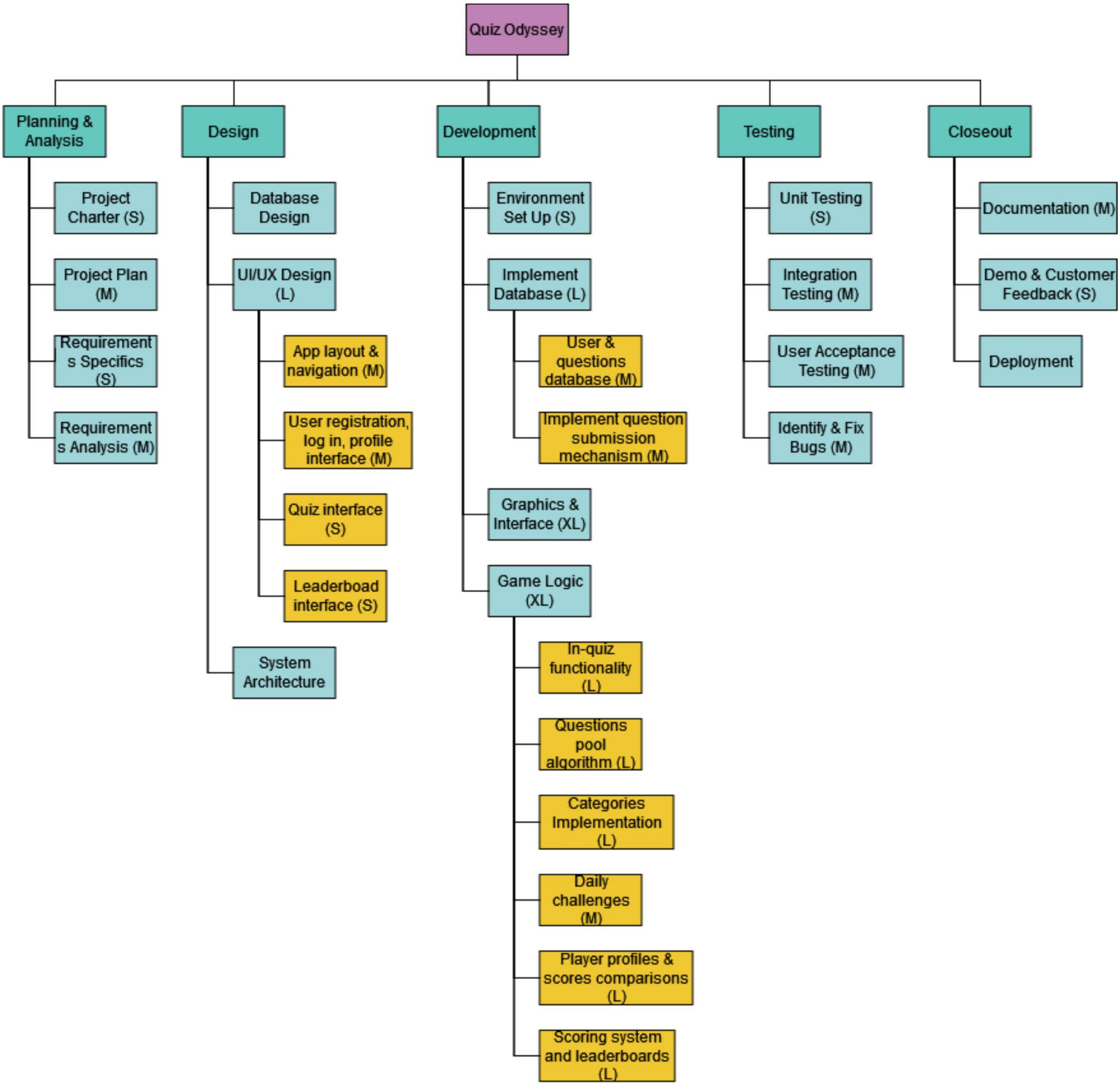
Non-functional Issues

1. Performance:
   - ○ Response Time: Define acceptable response times for user interactions, such as answering questions or loading the app.
   - ○ Scalability: Specify how the app should handle an increasing number of concurrent users.
   - ○ Load Testing: Conduct load testing to ensure the app can handle peak loads without performance degradation.
   - ○ Server Latency: Determine acceptable latency for data retrieval and server interactions.
2. Security:
   - ○ Data Encryption: Specify encryption standards for protecting user data during transmission and storage.
   - ○ Authentication: Define user authentication mechanisms, including password complexity and multi-factor authentication.
   - ○ Authorization: Specify access control mechanisms and user role-based permissions.
   - ○ Security Testing: Conduct security testing to identify and address vulnerabilities and threats.
   - ○ Compliance: Ensure the app complies with relevant data protection and privacy regulations, such as GDPR or HIPAA.
3. Usability and User Experience:

- ○ Accessibility: Ensure the app is accessible to users with disabilities, following WCAG guidelines.
- ○ User Interface Consistency: Define design standards for a consistent and intuitive user interface.
- ○ Responsiveness: Specify acceptable load times for app screens and transitions.
- ○ Error Handling: Define user-friendly error messages and user guidance for error situations.

Project Plan

Here is the WBS of the aimed project:


Quiz Odyssey
Planning & Analysis
Project Charter (S)
Project Plan (M)
Requirements Specifics (S)
Requirements Analysis (M)
Design
Database Design
UI/UX Design (L)
App layout & navigation (M)
User registration, log in, profile interface (M)
Quiz interface (S)
Leaderboad interface (S)
System Architecture
Development
Environment Set Up (S)
Implement Database (L)
User & questions database (M)
Implement question submission mechanism (M)
Graphics & Interface (XL)
Game Logic (XL)
In-quiz functionality (L)
Questions pool algorithm (L)
Categories Implementation (L)
Daily challenges (M)
Player profiles & scores comparisons (L)
Scoring system and leaderboards (L)
Testing
Unit Testing (S)
Integration Testing (M)
User Acceptance Testing (M)
Identify & Fix Bugs (M)
Closeout
Documentation (M)
Demo & Customer Feedback (S)
Deployment

Estimates

Activity	Man Weeks
Planning & Analysis	**22**
Project Charter (S)	5
Project Plan (M)	7
Requirements Specifics (S)	5
Requirements Analysis (M)	5
Design	**10**
UI/UX Design (L)	**6**
Design app layout & navigation (M)	2
Design user registration, log in, profile interface (M)	2
Quiz interface (S)	1
Leaderboard interface (S)	1
Database Design (L)	**2**
System Architecture (L)	**2**
Development	**18.5**
Environment Set Up (S)	**1**
Implement Database (L)	**5**
User & questions database (M)	2
Implement question submission mechanism (M)	2
Graphics & Interface (XL)	**3.5**
Game Logic (XL)	**9**
In-quiz functionality (L)	1
Questions pool algorithm (L)	2
Categories implementation (L)	2
Daily challenges (M)	1
Player profiles and score comparisons (L)	1
Scoring system and leaderboards (L)	2

Testing	**12**
Unit Testing (S)	5
Integration Testing (M)	2
User Acceptance Testing (M)	2
Identify & Fix Bugs (M)	3
Closeout	**6.3**
Documentation (M)	5
Demo & Customer Feedback (S)	1
Deployment (M)	0.3

Resources

Team Members

Name	**Student ID**	**Role**
Eren Taşdemir	150200035	Co-leader, full-stack developer
Berkant Bakışlı	150200069	Co-leader, full-stack developer
Karol Jan Charchut	912310003	Full-stack developer
Pijus Jonas Navasaitis	912310007	Full-stack developer
Egi Gjineci	150190910	Full-stack developer

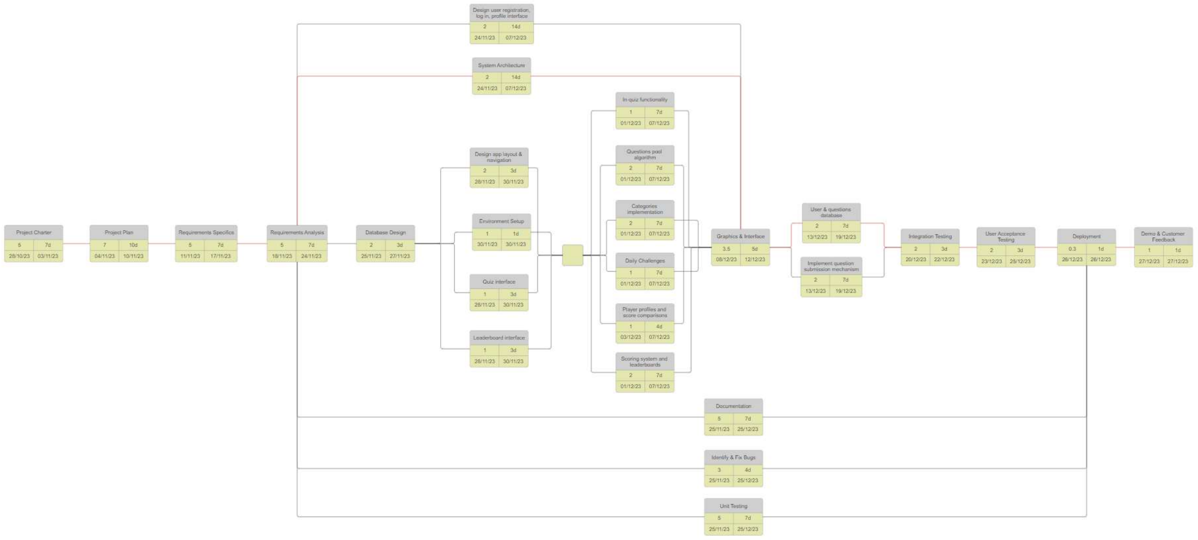
Task Assignments

Activity	**Assigned Members**
Planning & Analysis	
Project Charter (S)	Everyone
Project Plan (M)	Everyone
Requirements Specifics (S)	Everyone
Requirements Analysis (M)	Everyone
Design	

UI/UX Design (L)	Everyone
Design app layout & navigation (M)	Everyone
Design user registration, log in, profile interface (M)	Egi
Quiz interface (S)	Pijus, Berkant
Leaderboard interface (S)	Eren, Karol
Database Design (L)	Everyone
System Architecture (L)	Berkant
Development	
Environment Set Up (S)	Everyone
Implement Database (L)	
User & questions database (M)	Pijus, Karol
Implement question submission mechanism (M)	Egi, Karol
Graphics & Interface (XL)	Everyone
Game Logic (XL)	Everyone
In-quiz functionality (L)	Karol
Questions pool algorithm (L)	Pijus, Karol
Categories implementation (L)	Egi, Berkant
Daily challenges (M)	Pijus
Player profiles and score comparisons (L)	Berkant, Eren
Scoring system and leaderboards (L)	Eren, Egi
Testing	
Unit Testing (S)	Everyone
Integration Testing (M)	Everyone
User Acceptance Testing (M)	Everyone
Identify & Fix Bugs (M)	Everyone
Closeout	
Documentation (M)	Everyone
Demo & Customer Feedback (S)	Everyone
Deployment (M)	Berkant, Eren

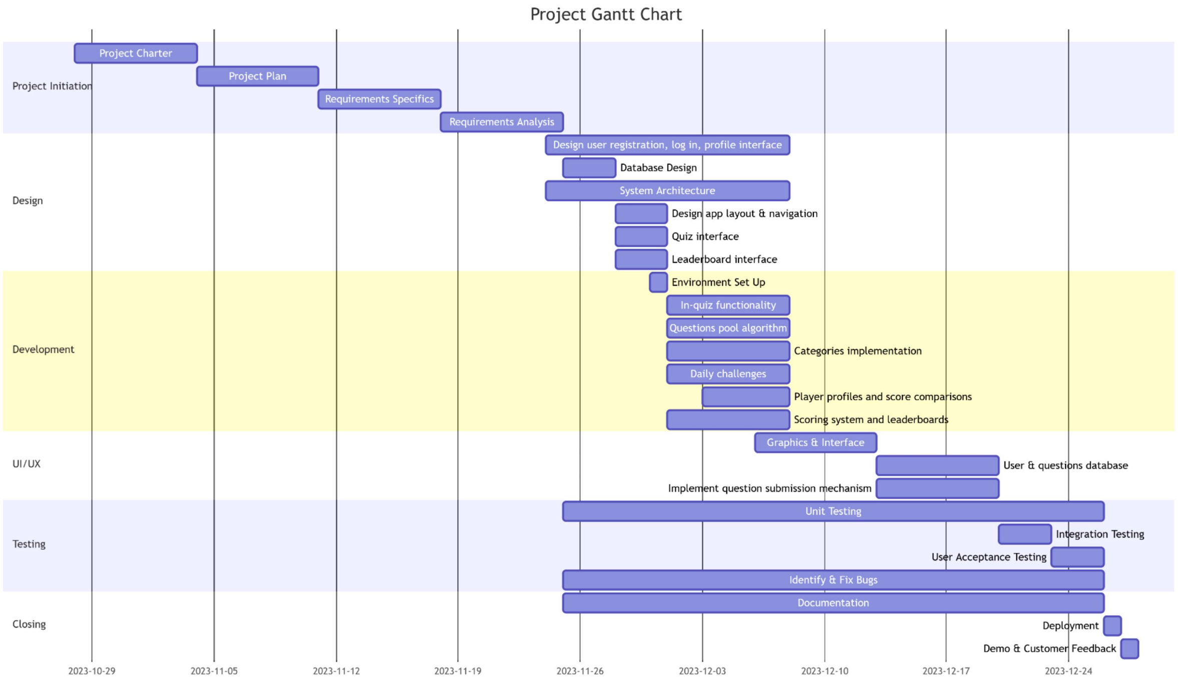
Schedule

PERT Chart



Critical Path is colored in red, and the schedule is written below each task in the form of start date and end date. Project is planned to be completed on 27/12/2023.

Gantt Chart



Risks

Risks	Category	Probability	Impact
Customer may add/change requirements	PS	50%	2
High staff turnover	ST	10%	3
Platform compatibility	TE	20%	2
Fewer users than estimated	BU	30%	1
Unforeseen delays	PS	50%	2
Cyber attacks	BU	5%	2
End of supports of technologies used	TE	10%	4
Developer team might be inexperienced in the tools	ST	80%	3

Risk Mitigation

Risks	Affect	Mitigation
Customer may add/change requirements	Project	Engage in regular communication with the customer to understand their needs and priorities, and educate them on the potential impact of scope changes on project timelines and costs.
High staff turnover	Project	Cross-train team members to ensure that knowledge is not concentrated in one person.
Platform compatibility	Product	Regular testing to address issues.
Less users than estimated	Business	Monitor and analyze user feedback.
Unforeseen delays	Project	Regularly update and review the risk register
Cyber attacks	Business/ Product	Address vulnerabilities, apply security best practices(e.g. minimize dependencies).
End of supports of technologies used	Project	Choose technologies that have long term support.
Developer team might be inexperienced in the tools	Project	Provide training and resources to the team. Consider mentorship.