Documentation

for

Arithmetic PvP

Prepared by

Kamil Agliullin, Dmitrii Shabalin, Evgeny Petrashko, Aidar Khuzin

Table of Contents

Requirements

1. Glossary

Introduction

Purpose

The glossary contains the working definitions for the Arithmetic PvP application. This glossary will be expanded throughout the life of the project.

Scope

This glossary addresses all terms which have specific meanings for this project.

Definitions

Tasks

Tasks are arithmetic expressions.

Game session(room)

Each game consists of tasks the player should solve. Game session is a place where one or more players are solving tasks.

Single player mode

The game mode where the player is playing neither cooperating nor competing with other players. Usually this mode is used to improve player skills.

Multiplayer mode

The game mode where players usually compete.

Single player levels

Level is the complexity of a game session. The higher level is, the more complicated tasks are used.

Rating system

When a user plays in multiplayer mode he/she receives points for wins and loses them for defeats. Based on this a player takes some place in the rating.

Invitation

A player can invite another player to join his game session. It could be done using some link or in-game requests.

In game request

A request received in the application interface. It could be used to accept a friend or accept an invitation.

Open room

A room that is accessible to everyone.

Private room

A room that is accessible only by invitation.

Progress bar for a player

The bar that shows how many tasks the player completed. So players can see how fast

others are doing.

In game currency(coins)
Coins that could be earned during gameplay. Possibly can be received using donations.

In game skins
Skins that can be purchased from the in-game shop.

Average speed
Average number of tasks solved per second.

2. Business Goals and Objectives

1	Make good-qualified product, which can be considered as a serious enough project for industry
2	Make the desired game interesting for the users
3	Make the gameplay comfortable for the users
4	Make secure game process, so rating in the game will be fair (no bots, no cheating)
5	Make useful product, so the users can improve their skills

3. Roles and responsibilities

Stakeholder's Name	Roles	Responsibilities
Evgeny Petrashko Dmitri Shabalin Aidar Khuzin Kamil Agliullin	Kamil - project manager, frontend developer, Dmitrii - frontend developer, Evgeny - backend developer, Aidar - backend developer	Kamil - project management, design template, deploy, security Dmitrii - web design(HTML, CSS, bootstrap) Evgeny - client side (vue.js) Aidar - server side (Django REST API)

4. Functional requirements

ID #	Features	Priority
1	Single player mode	Should
2	Multiplayer mode	Must
3	Levels in single player mode	Could

4	Rating system	Must
5	Join the game by invitation	Should
6	Invite people to the game	Should
7	Join the game in open rooms	Should
8	Authentication	Must
9	Statistics for multiplayer game	Must
10	Cross-platform web application	Must
11	Custom levels	Could
12	Statistics in single player mode	Should
13	In-game points	Should

5. User Stories

User Type	User Story Title	User stories
Web User	Single player	As a user, I want to have a single player mode, so I can train and improve myself
	Multiplayer	As a user, I want to have a multiplayer game, so I can compete with other people
	Levels	As a user, I want to have levels in single-player mode, so I can increase the complexity of the game process
	Rating system	As a user, I want to have rating system, so that I will know where I am in the world top
	Join the game	As a user, I want to have the possibility to join the game by invitation, so I can play with my friends
	Invite to the game	As a user, I want to have the possibility to invite people to my game, so I can compete with them
	Progress bar	As a user, I want to have the progress bar for the particular user in the game, so I can identify who is faster
	Open rooms	As a user, I want to be able to join games in open rooms so that I can compete with random people
	Authentication	As a user, I want to be authenticated in the game so that I can have my profile
	Statistics in multiplayer mode	As a user, I want to have a possibility to see my results after ending the game, so I can see my and others characteristics, such as spending time to the concrete question, average speed, total accuracy and place in the game top

	Security	As a user, I want to have a secure app, so other users can't use the 3rd-party programs and add points to themself
	Input without keyboard	As a mobile user, I want to have buttons with numbers, so that I can type not only with keyboard
	Skins	As a user, I want to have the possibility to buy skins using in-game currency(coins), so I can change the appearance of the site (design, progress bar etc.)
	Achievements	As a user, I want to have achievements, so I can show off them to my friends
	Cross-platform web app	As a user, I want to have a cross platform web app, so that I can access it from any device
	Chat	As a user, I want to have a chat, so I can be the part of a verbal counters community
	Multi-language	As a user, I want to have a multi-language app, so I can choose the appropriate language for me
	Custom levels	As a user, I want to have the possibility to customize the single-player mode, so I can choose the appropriate complexity for me
	Statistics in single player mode	As a user, I want to have a possibility to see my results after ending the single player game, so I can analyze my statistics, such as average speed, total accuracy and place in the world top for this level
	Points	As a user, I want to have the possibility to get points, so I can promote myself in the world top

6. Non-Functional Requirements

Requirement	User Story	How will you achieve it
Performance	As a user I want to have fast response from the server, so I can rate my speed without considering network delay	We will meet this requirement by generating the questions before the game starts, so the delay will be minimized.
Security	As a user I want to have third-party programs/bots blocked, so the game is fair.	We will meet this requirement by creating a qualified API, so there are no API calls as "add me 100 points". Also, we will send math expressions to users in image format, so if somebody (bot) steals this question, he spends a significant amount of time getting the actual question.
Portability	As a user I want to play the game from the mobile devices too, so I can play on the go.	We will meet this requirement by creating the adaptive templates, so it will scale from one device to another.
Ease of use	As a user I want to have	We will meet this requirement by following the Material Design, so the interface will be simple and laconic.

	simple interface, so I easily can find what I want	
Reliability	As a administrator I want a reliable game, so malicious user can't put the server down with too many requests	We will meet this requirement by restricting the amount of requests per second/per minute/per hour from the users.

7. Software Development plan

Inception Phase				
#Iteration	**Timeline**	**Stakeholders**	**Activities**	**Artifacts**
#1	24/08/2021-25/08/2021	all	Determine Business goals and objectives with valid justification Identify the stakeholders Establish roles and responsibilities	Deliver the documentation of achieved milestones
#2	25/08/2021-26/08/2021	all	Requirement engineering(20% user stories) Identify Risks	Update the documentation of achieved milestones with User stories

Elaboration Phase				
#Iteration	**Timeline**	**Stakeholders**	**Activities**	**Artifacts**
#1	25/08/2021-25/08/2021	all	Revise User Stories (100%)	Document 100% user stories
#2	26/08/2021-26/08/2021	all	Software development planning	Iteration Plan
#3	27/08/2021-27/08/2021	all	Software Architecture Test Plan	Software architecture document Test Plan Document

Construction Phase				
#Iteration	**Timeline**	**Stakeholders**	**Activities**	**Artifacts**
#1	28/08/2021-10/09/2021	all	Implement Feature 2	Github repository Merge branch for the feature Unit testing result
#2	10/09/2021-12/09/2021	all	Implement Features 5, 6 Unit test cases for features 5, 6	Github repository Merge branch for the features Unit testing result

#3	13/09/2021 - 17/09/2021	all	Implement Features 7, 8, 10, 4 Unit test for features 10, 4	Github repository Merge branch for the features Unit testing result
#4	18/09/2021 - 24/09/2021	all	Implement Features 1, 3, 17, 18, 19 Unit test for features 19	Github repository Merge branch for the features
#5	25/09/2021 - 27/10/2021	all	Implement Features 12, 15, 9, 11	Github repository Merge branch for the features
#6	28/09/2021 - 05/10/2021	all	Implement Features 13, 14, 16, 20 Unit test for features 20	Github repository Merge branch for the features Unit testing result

Transition Phase				
#Iteration	Timeline	Stakeholders	Activities	Artifacts
#1	5/10/2021 - 8/10/2021	all	Integration, End to end testing Training for Users and Developers	Github repository Merged branches Integration and ended to end test results Final README for developers and Users
#2	9/10/2021	all	Final product release	Working Product