RUP ARTIFACTS FOR Innogl

Prepared by:
Vladislav Lamzenkov, Maxim Ksenofontov, Mohammad Khalil

03.10.2021

Version History			
Editor's Name	**Date** (DD/MM/YYYY)	**Reason for Changes/Sections Updated**	**Version**
Innogl	01/09/2021	Update user stories	1.0.1
Innogl	03/09/2021	Update business goals and software development plan	1.0.2
Innogl	20/09/2021	Updating business goals, user stories and non-functional requirements	1.0.3
Innogl	25/09/2021	Updating design documentation	1.0.4
Innogl	30/09/2021	Updated links	1.1.0

Table of Contents

1. Project Information

This project is an anonymous web and video chat with strangers. Users can choose an online companion to discuss anything, find new friends or discuss their problems. Besides, if they agree, they can turn on their cameras to see each other leaving disregarding anonymity. Users are able to add a specific topic which they want to discuss and wait for someone to choose their own topic, on the other hand, they can take someone's topic to chat about in case they don't have specific ideas. If users don't prefer the previous options, they may choose a chat room randomly without a specific topic.

2. Business Goals and Objectives

- Improve communicative skills of people with strangers.
- Help foreign students to learn English.
- User-friendly website.
- Well-organized content.

3. Roles and responsibilities

Stakeholders' Name	Roles	Responsibilities
Project manager	SCRUM master	Controlling the flow of development, planning and time-management, supports the development team by prioritizing the product backlog, be sure that deadlines are met.
Customer	User	Use the application for anonymous chatting, giving the feedback anonymously.
Developer	Front-end developer, QA Engineer, back-end developer, DevOps Engineer	Developing the front-end part of the web application, testing the code and checking the quality of the product, finding bugs, developing the back-end part of the web application on Spring(Java), deploying the application on the remote server, specify the hardware requirements for server based on the project requirements.

4. Requirement Analysis and Specifications

4.1 Features

ID #	Feature	Priority	Details
1.1	No user registration	Must	No registration, anonymous users
2.1	Send private messages	Must	Personal one-to-one chats only
2.2	Make a video call	Preferable	Personal video calls
2.3	Topic-Separated Chat Rooms	Preferable	Connect two people based on their shared interest

4.2 User Stories

User Type	User Story Title	User stories
Web User	Registration	1. As a user, I am able to use the application anonymously (without registration).
	Dashboard	1. As a user, I want to video call my partner from my chat thread so that I interact with them on video on the internet. 2. As a user, I want to increase or decrease the font size of the text in chat so that I can set the font size appropriately. 3. As a user, I want to see how many people are online on the site, so that I'll be aware of the differences in chats.

		4. As a user, I want a skip icon in the chat so that I can move from the current chat to another one.
	Privacy and support	1. As a user I want to maintain privacy about the chat so no one is able to know what was the conversation and with whom. 2. As a user, I want to read **Innogl** terms of service and privacy policy before registering so that I am aware of the app's services and policies. 3. As a website visitor, I want to talk to support teams so that I can get my questions answered quickly.
	Services	1. As an international student, I want to speak a lot with people in English so that we can understand each other. 2. As a user don't want to stack and wait for a new chat room so I'm supposed to receive a notification when I got a chat partner.

4.3 Non-Functional Requirement

NFR	Sub-Characteristics	How will you achieve it
Usability	Adaptivity	- **The app should work from PC, tablets and mobile phones.** - To reach it we will use React with CSS grid to make an adaptive website.
Performance	Capacity	- **The servers must be able to handle 1000 requests per second.** - To reach it Kubernetes will be used to scale the app horizontally.
Reliability	Accessibility	- **The website should be available even if the back-end part is down**

		- We separate the front-end part from the back-end part so that the client interface will be almost always accessible, and server errors will be handled on the client side.
Data Integrity	Database	- **The project should have a database with fast access to be able to store information available sessions.** - We will use Redis as an in-memory database for fast I/O operations.

5. Software Development plan

Inception Phase				
#Iteration	**Timeline**	**Stakeholders**	**Activities**	**Artifacts**
#1	24/08/2021-27/08/2021	Maxim Ksenofontov, Vladislav Lamzenkov, Mohammad Khalil	Find an idea for the project Determine Business goals and objectives with valid justification Identify the stakeholders Establish roles and responsibilities	Deliver the documentation of achieved milestones
#2	28/08/2021 - 31/09/2021	Maxim Ksenofontov, Vladislav Lamzenkov, Mohammad Khalil	Requirement engineering(20% user stories) Preparation of the project environment Determine project feasibility and risks	Update the documentation of achieved milestones with User stories Github repository Risk list

Elaboration Phase				
#Iteration	**Timeline**	**Stakeholders**	**Activities**	**Artifacts**
#1	01/09/2021 - 03/09/2021	Maxim Ksenofontov, Vladislav Lamzenkov, Mohammad Khalil	Requirement engineering(100% user stories) Revise User Stories (100%) Verify project requirements according to the architecture A stack of technologies to the skeleton architecture Design, implement, and test a skeleton structure of the system by designing, and testing a small number of critical scenarios	Document 100% user stories Some critical scenarios when the project may lead to the unpredicted result(possible bugs) A skeleton of the architecture with the stack of technologies
#2	04/09/2021-06/09/2021	Maxim Ksenofontov, Vladislav Lamzenkov, Mohammad Khalil	Build a full software Architecture based on the skeleton Produce an accurate schedule Software development planning	Iteration Plan RUP artifact
#3	07/09/2021 - 12/09/2021	Maxim Ksenofontov, Vladislav	Apply user scenarios to create a test plan(unit testing, integration testing)	Software architecture document

		Lamzenkov, Mohammad Khalil	Choose a style of development	Test Plan Document

Construction Phase				
#Iteration	**Timeline**	**Stakeholders**	**Activities**	**Artifacts**
#1	13/09/2021 - 20/09/2021	Maxim Ksenofontov, Vladislav Lamzenkov, Mohammad Khalil	Implement Feature 1.1 Unit test cases for Feature 1.1 Implement Feature 2.1 Unit test cases for Feature 2.1	Working Feature 1.1 branch Unit test results for Feature 1.1 Working Feature 2.1 branch Unit test results for Feature 2.1
#2	21/09/2021 - 27/09/2021	Maxim Ksenofontov, Vladislav Lamzenkov, Mohammad Khalil	Implement Feature 2.2 Unit test cases for Feature 2.2 Implement Feature 2.3 Unit test cases for Feature 2.3	Working Feature 2.2 branch Unit test results for Feature 2.2 Working Feature 2.3 branch Unit test results for Feature 2.3

Transition Phase				
#Iteration	**Timeline**	**Stakeholders**	**Activities**	**Artifacts**
#1	28/09/2021 - 02/10/2021	Maxim Ksenofontov, Vladislav Lamzenkov, Mohammad Khalil	Revise code and refactoring Revise documentation for the code Possible deploying Integration testing End to end testing Training for Users and Developers	Github repository Documentation for the project Merged branches Integration and ended to end test results Final README for developers and Users Possible working application deployed on the remote server(cloud)
#2	03/10/2021 - 03/10/2021	Maxim Ksenofontov, Vladislav Lamzenkov, Mohammad Khalil	Final product release	Working Product

6. Useful Links

- Github repository: https://github.com/e2xen/innogl
- The link to the video with the demo is available in *README.md* of the repository: https://github.com/e2xen/innogl
- Design documentation is available in *README.md* of the repository: https://github.com/e2xen/innogl