Project Documentation

ISTE-330 Database Connectivity and Access

Chat App

**Document (filename):**

**Author(s):**

Filip Gadžo

Luka Zlatarić

Ivan Marušić

Ljubo Zadro

Antonio Šeparović

**Date (last change): 21/04/2020**

**Version (last version): 21/04/2020**

|  |  |  |  |
| --- | --- | --- | --- |
| *Version* | *Description of Change* | *Author(s)* | *Date* |
| 1.0 | M0 |  | 2020-03-20 |
| 1.1 | M1 | Aš,LJZ,LZ | 2020-04-07 |
| 2.0 | M2,M3,M4,M5 | FG, IM, LZ | 2020-04-21 |

Table of contents

[1 Introduction 3](#_heading=h.30j0zll)

[1.1 Overview 3](#_heading=h.1fob9te)

[1.2 Purpose and Scope 3](#_heading=h.3znysh7)

[1.3 Background 3](#_heading=h.2et92p0)

[1.4 References 3](#_heading=h.tyjcwt)

[1.5 Document Overview 3](#_heading=h.3dy6vkm)

[2 Problem Description and Solution Architecture 4](#_heading=h.1t3h5sf)

[2.1 Problem Description 4](#_heading=h.4d34og8)

[2.2 Technologies and Architectural Design 4](#_heading=h.2s8eyo1)

[2.3 Database Layer and Database Connectivity Layer 4](#_heading=h.17dp8vu)

[2.4 Business Layer 4](#_heading=h.3rdcrjn)

[2.5 Presentation Layer 5](#_heading=h.26in1rg)

[2.6 Areas of particular concern 5](#_heading=h.lnxbz9)

[3 Requirements 6](#_heading=h.35nkun2)

[3.1 Context 6](#_heading=h.1ksv4uv)

[3.2 Functional Requirements 6](#_heading=h.44sinio)

[3.3 Other (Non-Functional) Requirements 6](#_heading=h.2jxsxqh)

[4 User Documentation 7](#_heading=h.z337ya)

[4.1 Graphical User Interface Design 7](#_heading=h.3j2qqm3)

[4.2 User Manual 7](#_heading=h.1y810tw)

[5 Installation, Configuration and Acceptance Testing 8](#_heading=h.4i7ojhp)

[5.1 Installation 8](#_heading=h.2xcytpi)

[5.2 Configuration 8](#_heading=h.1ci93xb)

[5.3 Acceptance Testing 8](#_heading=h.3whwml4)

[6 Final Remarks and Conclusion 9](#_heading=h.2bn6wsx)

# Introduction

## Overview

*[Provide an overview of the project and application (that will be developed) and any additional information to place it in the context. Make a high-level description of the business domain and the problem your project and application will solve (and how). Provide a high-level description of the desired functionalities that will be later described in detail. Several sentences are expected (****deliverable and milestone #0 and #1****).]*

The project will be about a chat application. Hopefully it will be connected to RIT student accounts. Students will be able to chat between themselves but will also be able to create group chats with different user rights. Between groups users with more privileges could be able to assign tasks to other users. So basically the project is about a chat application with some features of the task management system.

## Purpose and Scope

*[Provide an overall description of this document and the project. Explain the purpose of this project and the intended clients. Additionally, discuss the scope of this document (what it describes in the context of the project and also what it will not describe) as well as the intended audience of this document (managers, developers, clients, etc.). Several sentences are expected (****deliverable and milestone #0 and #1****).]*

Application will be a chat app with RIT login system. It will also have the feature of the task manager. Application will serve the purpose of chatting with one user or even creating a group where multiple users can chat. The difference from all the other applications is that the admin of the group can give tasks to other group members, and members who got the tasks can report to the admin about the progress of that task, but the admin will also be able to track every task by himself. Also, the admin would be the owner of the group, and will be able to assign different positions, like co-owner, admin, member, and many more and each role will have different excess rights.

## Background

*[Describe who is producing this document and why – you can say a few sentences about your team and why you are doing it. Several sentences are expected (****deliverable and milestone #0 and #1****).]*

The people producing this document are 5 IT students from Rochester Institute of Technology (RIT). The reasons why we are producing such a document is so as to better plan and understand our project. To get a better insight into what our project will consist of, and how it will be implemented. This document servers as a guideline into how the programmers will properly construct, create, and manage such an application. To produce results in a quick and effective way. It’s also intended for the project’s supervisor to better inform him/her about the project’s architecture and logic. To get a better view into what the programmers are building.

The programmers producing this project are highly skilled in their field, their job is to learn and implement new ideas and gain experience. The whole purpose of this project is to help us better understand what goes into creating a project, and the skills needed to implement one. Many

obstacles will be presented to us in time, and so it’s better to be accustomed to the working environment. We communicate and work as a group and we intend to finish our work on time.

## References

*[List all necessary references and other associated documents, including links to any documentation (about programming languages, database, drivers, frameworks, libraries, tools, and similar software artifacts), used standards, whitepapers and policies, formats of deliverables you are going to produce, etc. Several sentences and lists with links are expected (****deliverable and milestone #0 and #1****).]*

MySQL - database layer, where all of our data is stored. <https://www.mysql.com/>

Spring boot - it is a framework which we will use to create our web based java application <https://spring.io/>

JWT authentication - json web token will be used for authenticating users . After validating user every user get its generated token, and client will send this token inside authentication header for every request he/she make inside an app. <https://jwt.io/introduction/>

ReactJS - it is a javascript framework, it will be used for creating front-end of our application together with css and other styling components that go with it. <https://reactjs.org/>

<https://spring.io/blog/2015/09/01/react-js-and-spring-data-rest-part-1-basic-features> - this is the link that all of us got through for purposes of learning how to create web application with react and spring.

## Document Overview

*[Provide a short description of the document organization per chapter – what is described in each chapter. Several sentences are expected (****deliverable and milestone #0 and #1****).]*

In the first chapter it is all about introduction,what project is all about and who is behind it. In the second chapter there are a bit more technical things.Firstly there is problem description.In that part there are information about what should be developed and how it will be developed.In that part of document we are also presenting database.We are presenting different layers and how each layer will develop over time.At the top end of that part there are information about some challenges that will we pass through while developing this application.

In the requirements part, we will provide info about the functional and other requirements of this project. Finally, at the end of the document, there will be useful information for the users. You can find there information such as how to build and start the project.

# *.*Problem Description and Solution Architecture

*[This chapter should be started for the* ***deliverable and milestone #0****, and some parts (problem description and technologies and architectural design) should be finished for the* ***deliverable and milestone #1****. However, some parts of it will be produced and delivered in* ***deliverables and milestones #2, #3, and #4****, but no later than the final* ***deliverable and milestone #5****.]*

## Problem Description

*[This should be a detailed description of the problem that will be solved. Explain what a problem that you are trying to solve is (in detail), are there any other existing solutions, and how will your solution be in comparison to those (same or better). This should go much more into details than in the previous chapter. At least several sentences are expected (****deliverable and milestone #0 and #1****).]*

The problem we as a team are trying to solve is the problem that we as

a team of college students are facing during the development phase.

Everyday we use applications like whats'app, github, mycourses, some of

us may use slack if they work. Idea is to have essence of all these technologies

combined. A team of developers should be able to communicate without any

problems no matter if they are positioned all in one office or everyone

works remotely from the comfort of their home. And if there is one app that

should be imitated for that is whatsApp. To be able to chat and

having conversations with each other is really important. But what

whatsApp lacks is to have better organizations by groups and similar.

So we want for admins and team leaders to be able to easily create

groups and members for different projects or any other purposes.

With these chat and grouping organizations we covered only one part of the

app, the second part is task manager. We want to be able to assign

developers to their tasks and be able to track in which stage is the

task. For example we have three stages, in progress and done.

## Technologies and Architectural Design

*[****IMPORTANT:*** *Capture important software design decisions and describe them – why did you decide to do it this way and what technologies are you going to use. Do not forget to explain the choice of programming language(s) and layers, and rationale for this design of your application. Describe and sketch Software Architecture (picture) – decomposition of the software into layers/modules/units/components and provide detailed software design description for each part (layer). At least several sentences and an architectural design figure (with layers and technologies) with description are expected (****deliverable and milestone #0 and #1****). ]*

For this project, we decided to use REST (Representational State Transfer) and the main reason for considering it is that we wanted the backend to be independent of the frontend part.

For the frontend part, we decided to use a JavaScript framework React.js.Not only that React is one of the most popular tools nowadays to build web applications but its concept of components will perfectly fit our app.

For the backend part, we decided to use Java. We choose Java because Java’s WebSocket will be perfect to use in our chat application. We already have some experience with Java WebSocket and we think that it will be a good choice.

Most likely we will use the MySQL database for storing data from our application. We think that it will be good enough for something like this.

## Database Layer and Database Connectivity Layer

*[****IMPORTANT:*** *Provide database structure with ERD and database schema as well as data dictionary explanations of entities (tables) and their characteristics/attributes/properties (columns). At least several sentences are expected to describe the model. There should be a physical database model figure for the chosen database with 10-15 tables in a clear relationship, with a description of each database table. The data dictionary is more than welcome – you can do it in the usual table format in Word, where each row represents one attribute from a database table, and you should have it for all attributes and all database tables. Each row should hold information about that attribute, including table name, column name, the data type in the database, short description, constraints (e.g., primary or foreign key) and restrictions (format of the data), and (if unclear) sample data. You should start doing it for the* ***deliverable and milestone #1*** *and finish it no later than* ***deliverable and milestone #2****).]*

USER- user table has userID as its primary key. Last name and first name are other 2 attributes inside that table.

UserMedia - this is where data like user profile photo and similar things are stored. mediaID is Pk and userID is FK

GroupMembers - many to many relationship with User table and one to many with Groups table. memberID is PK and userID is FK. memberID is FK inside Groups table. We add members for every group.

Groups - group table have groupID which is id for some particular group. Every group has members which are assigned their memberID and that is FK from GroupMembers table. Every group have their name and number of members.

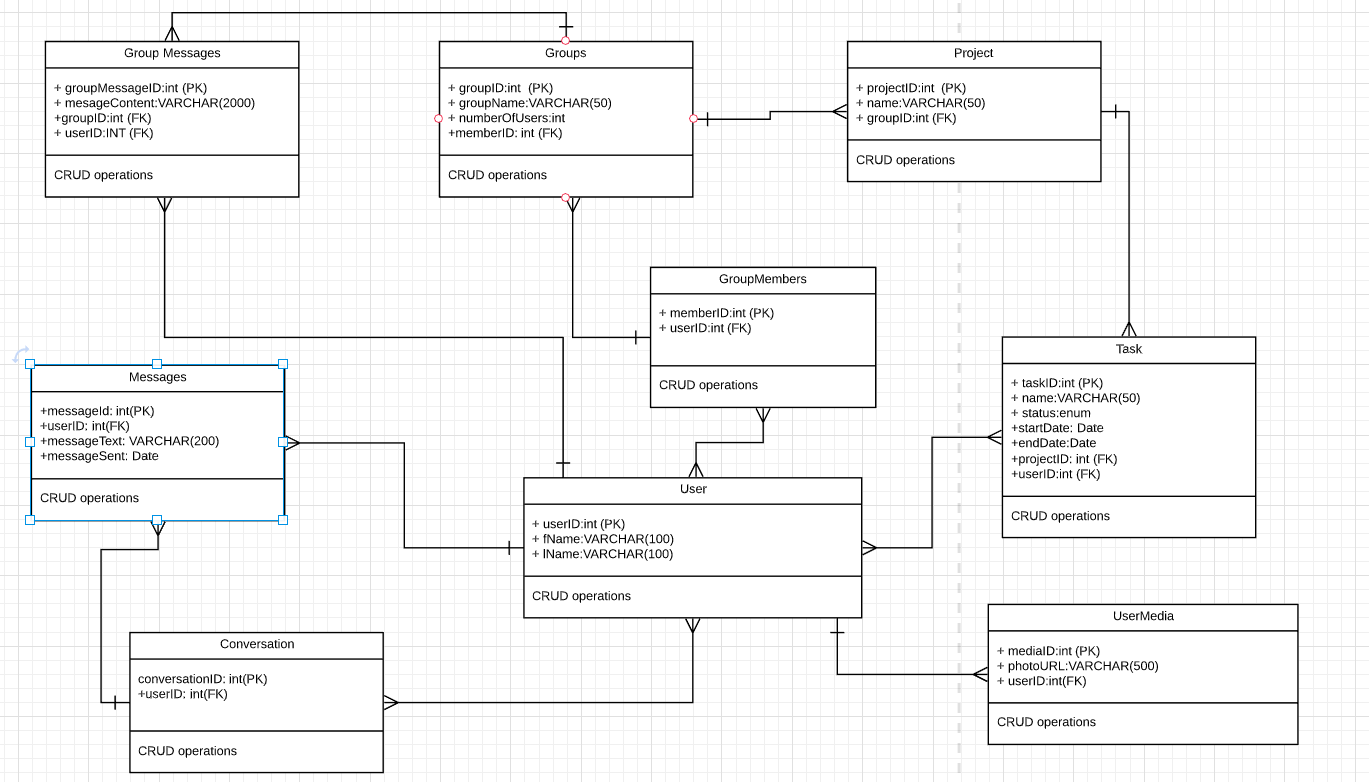
Project - every project has its own projectID which is PK of this table. Every project has name as attribute and group which is assigned to that project. That is why groupID is FK in this table.

Task - Inside every project there are tasks assigned either to group or to single user. Every task has its own unique taskID which is PK, name of task, status (can be TODO/INPROGRESS/DONE), date when task is assigned and time when task is expected to be done. ProjectID and userID are FK’s to this table.

Messages - this table represents messages which is one of main features of this app. This table store messageID which is PK, userID which is FK, messageContent and time stamp in format of date, messageSent.

Conversation - conversationID PK, userID FK. This table stores individual 1v1 conversations.

GroupMessages - Same grous from Group table have their own privete chat room. groupMessageID is PK inside of this table . messageContent - content of message. groupID and userID are FK’s in this table.



## Business Layer

*[Provide Business Layer description, design and explanation, as well as describe connections to Database Connectivity Layer and Presentation Layer. At least several sentences are expected to describe how the Business Layer will be constructed, what its purpose is, and how it communicates with Database Connectivity Layer (below) and Presentation Layer (above). You should start doing it for* ***deliverables and milestones #1 or #2*** *and finish it no later than* ***deliverable and milestone #3****). ]*

We are using spring boot framework for creating our business database layer. JDBC will connect mysql with our app. Business logic will be inside controller and repository packages. Repository package is an abstraction layer for all of our CRUD operations. Controller is where actually most of the business logic is and it will communicate with presentation layer.Each table in the database will monitor one class in Spring.Each class will have CRUD methods as well as getters and setters for all attributes.We will also have one class that will be responsible for creating and opening connection with the database.

## Presentation Layer

*[Provide Presentation Layer description, graphical user interface (GUI) design, including structure, layout and explanations, as well as a description of used technologies. At least several sentences are expected to describe how the Presentation Layer will be constructed, what its purpose is, and how it communicates with the Business Layer (below) and the users (clients). You can also include all possible actions, menus, and options. You should start by doing some prototypes or wireframes for* ***deliverables and milestones #1 to #3*** *and finish it no later than* ***deliverable and milestone #4****. In the final* ***deliverable #5*** *you can even provide some screenshots.*

The presentation Layer will be GUI(Graphical User Interface), and it will be the only layer that is visible to the user.GUI will be developed using ReactJS(JavaScript framework). React is based on components that together are one cohesive unit. We will create different components for different parts of the GUI. For example, one component will be the sidebar where all users will be listed, another component will be a container where the user will enter the message, etc. For the overall design, we will use React’s design framework MaterialUI.We will communicate with the presentation layer by calling functions that will be responsible for that. From GUI data will be read from the database as well as data will be sent to the database. Users will be able to interact with the presentation layer. Some examples of those interactions are: sending a message to another user, creating a group of users, assigning a task to another user, etc.

## Areas of particular concern

*[In this chapter, you should provide identification of areas of particular note or concerns. It could be about prerequisites (which must be respected) and assumptions, as well as possible risks for your project. Those could be related to an organization, planning, resources, technologies, and availability, as well as team members. You can describe a plan on how to mitigate those risks. You should start doing it for* ***deliverables and milestones #1 to #3*** *and finish it no later than* ***deliverable and milestone #4****.]*

There are some prerequisites for this project to be done. First of all, some team members must learn some technologies that will be used in the project. For example, ⅖ team members have good knowledge of ReactJS, and for all of us, this is kinda first touch with Spring. Besides that, we are now separated so we have to work via some online tools, and in the working environment like that, it is hard to look for progress, to look how each team member is working, etc. In our opinion, those 2 challenges are the biggest risks for this project, and if we get over them we will definitely finish this project on time.

# Requirements

*[This chapter should be started for the* ***deliverable and milestone #0****, and some parts (context and functional requirements) should be finished for the* ***deliverable and milestone #1****. However, some parts of it will be produced and/or changed later in* ***deliverables and milestones #2, #3, and #4****.]*

## Context

*[Provide a description of the application in the broader context, how it will work within the environment of other systems (e.g., payment systems if there is some kind u purchasing involved), with explanations as applicable. The context of a system refers to the connections and relationships between the application and its environment. At least several sentences are expected and it relates to* ***deliverables and milestones from #1 to #5****.]*

As the topic suggests our project will be based on a simple chat like system e.g. Whatsapp. It will require internet connection and also connection to the database layer. Like any chat application our app will require a login system so that the users can safely and securely enter and use their accounts. Some users will have higher privileges than ordinary users(programmers, app owner,...). To apply this feature means we will need to keep a log of the users account and password. Once the user is logged in he/she can communicate individually with other users or form a group with many users. As an extra feature we will also provide in the group chat a task feature option. Just like in Slack, Asana, Github label tasks,... we can provide a task object to the user where he/she can update and notify the task manager on where that individual tasks progress is standing. With this much features in our application we will need to hold a variety of data from history messages, tasks, friends, groups, accounst.

## 

## Functional Requirements

*[****IMPORTANT:*** *List, name and explain all key functionalities – there should be approx. 15 functional requirements listed. You should provide a table where each requirement is named (short code or name, could combine letters with numbers) and a detailed description, as well as who is responsible or performing the action associated with this functional requirement. You could also use use-cases (diagrams or descriptions), including use-case names, actors, events flow, exceptions, and special requirements. Include user requirements if necessary (users, roles, privileges) and associate with specific functional requirements. This should be started for the* ***deliverable and milestone #0****, and the proposal must be agreed upon and finished with* ***deliverable and milestone #1****. Later, with the approval of the client, it could be revised in* ***deliverables and milestones #2 - #4****).*

|  |  |
| --- | --- |
| Functionalities | Description |
| Log In system | For using our application users would need to provide username and password which will be used for handling accounts. While we develop our application we will use usernames and passwords provided by us, but in future we would like to work with the RIT account system. Accounts will be divided in admin, member, developer, professor,...  Now the authentication is done with jwt. |
| Admin | Users with admin privileges will be able to monitor work of the whole group, giving tasks to other members , get reports from all members from the group and communicate with members . |
| Member | Users with member privileges will be able to communicate to other members of a group, to send reports to the admin and work on given tasks. |
| Professor | Users with professor wrights will be able to monitor all actions of a group and communicate with them in separate chat. |
| Account settings | Every user will be able to manage settings such as adding profile picture, personal data and change his account name. |
| Dark mode | Every user will be able to choose between light or dark background of the application. |
| Group chat | This feature will be automatically generated when a group is made. All users except the professor will be able to see messages and contribute in a conversation. |
| Professor chat | Every group will have a separate chat with the professor so they can always express their concerns or questions about certain tasks. Professor can also send his feedback about tasks that are done , warn some students to work more or send kudos. |
| Task list | Every user except professor(because he wont have any) will be able to see their tasks with comments or explanations listed inside of application. |
| Task managing | Every user except professor(because he wont have any) will be able to set feedback on a task such as start/on going/done. |
| Task prioritizing | Admin will be able to set priority on each task such as urgent/primary/secondary... |
| Github connection\* | As we progress with our application we would like to connect it to github so that repositories are connected to a certain group. |

## Other (Non-Functional) Requirements

*[Describe the non-behavioral and non-functional requirements, including hardware and software requirements (e.g., platforms needed to support this application), programming interfaces, and any operational requirements (how the system will run and communicate with the environment). You could also provide information about application availability (time of day or week), general performance (how fast it should be in client responses), capacity (how many concurrent users or connections it will support), error handling (how is it handled), conventions used, security and similar if necessary. This should be started for the* ***deliverable and milestone #1****, and it could be revised in* ***deliverables and milestones #2 - #5****).]*

Platforms that you need to have to start an application are mysql server, mysql workbench, any IDE for starting java applications and npm.

# User Documentation

*[Usually, this chapter should be started later, and at least partially filled for the* ***deliverable and milestone #3****, and should be finished with* ***deliverable and milestone #4****. However, some parts of can be changed later, even in* ***deliverable and milestone #5****.]*

## Graphical User Interface Design

*[You should provide user design and user experience description, as well as a description of used technologies. This should be started for the* ***deliverable and milestone #3****, and should be finished with* ***deliverable and milestone #4****.]*

User is first introduced to a login page where he needs to provide username and password. In case that the username is new to the application he can create a new account. After successful registration he can login and he is brought to the homepage of the application. There he can see on the right all the users and groups he is connected to and co-operating with. In the center of the app is the chat box where he can communicate with other users. On the right, users can see their tasks that are provided to them and will be able to add new tasks and delete tasks that are being done.

## User Manual

*[This should provide expected usage of the available functionalities, could be divided per user roles, and should include screenshots with detailed descriptions. This should be started for the* ***deliverable and milestone #3****, and should be finished with* ***deliverable and milestone #4****.]*

Admin of the application is permitted to do whatever with the application. He is able to create groups, kick or remove users from the application and also assign specific tasks to them. He will be able to see all the logs and all the changes that are being made.  
Users will be able to only see their groups, other friends and tasks they need to do.

# Installation, Configuration and Acceptance Testing

*[Usually, this chapter should be started later, and at least partially filled with* ***deliverable and milestone #3 or #4****, and should be finished with* ***deliverable and milestone #5****.]*

## Installation

*[Provide a technical manual – prerequisites and installation process description details. Should be finished with* ***deliverable and milestone #5****.]*

To install this project to make it runnable and usable to the user then firstly the user must get the entire project from the vendor. We have implemented this process by loading from our github or bitbucket group account. This file will be provided to the user. After having the file, the user must open his/her terminal/CommandLine and selects the projects directory. Once the project directory is selected then install npm. Once installation is complete type npm start and it will start executing and running the project program.

## Configuration

*[Technical manual should also hold configuration detail and default values for this project to work. Should be finished with* ***deliverable and milestone #5****.]*

To run our project you need to go through several steps:

1. open mysql work bench and run this script:

create database chattask;

use chattask;

select \* from users;

INSERT INTO roles(name) VALUES('ROLE\_USER');

INSERT INTO roles(name) VALUES('ROLE\_ADMIN');

2. Open backend project and go to src/Main/resources and open application properties file

spring.datasource.url= jdbc:mysql://localhost:3306/chattask

spring.datasource.username= root

spring.datasource.password=

inside provide information about your database

3. Run backend project

4. Open frontend folder in cmd and write these commands:

npm install

npm start

\* you need first to sign in to enter application

## Acceptance Testing

*[Some acceptance testing should be performed to determine if the requirements are met – you should describe typical usage and tests to be executed for the application. Should be finished with* ***deliverable and milestone #5****.]*

After you successfully signed in you will be prompted that your sign in was successful. After that you should be able to log in and start using the application.

# Final Remarks and Conclusion

*[Usually, this chapter should be started later, and at least partially filled with* ***deliverable and milestone #3 or #4****, and should be finished with* ***deliverable and milestone #5****. You should summarize the experiences, both in terms of the produced results and work on the project. List all project deliverables, as well as positive (and negative) experiences and concerns. Comment on missing functionalities and possibilities for improvement and extensions. Estimate project effort (person-hours) and how it was distributed in time and per team roles. This chapter can also include a work log summary for all team members (for each day who did what).]*

While developing this project we have experienced many ups and downs. Many of us have been faced with difficult challenges and deadlines. We developed this project with immense difficulties and hurdles. Nonetheless, we have accomplished our goal the best we could and produced a project as closely as we could to the desired project goal. Work on the project was very much time consuming and difficult to implement.