**Functional design Waifu Logics**

**Project: The I.T. Connection**

**Client: Waifu Logics**

**Projectnumber: 001**

**Auteur: Guylian Gilsing, Peter Janssen, Duncan Sterken**

**Date: 23-4-2018**

**Version: v0.0.1**

The undersigned declare their agreement with the content of this functional design.

**Client Projectmanager**

***Initial Seen: Initial Seen:***

Date: <Geef de datum op.> Date: <Geef de datum op.>

Place: <Geef de plaats op.> Place: <Geef de plaats op.>

Functional design document

Inhoud

[PROJECT MANAGEMENT 2](#_Toc498596712)

[PROJECT DETAILS 2](#_Toc498596713)

[PROJECT DESCRIPTION 2](#_Toc498596714)

[REQUIREMENTS 2](#_Toc498596715)

[CONTENT OF A REQUIREMENTS DOCUMENT 3](#_Toc498596716)

[USE CASE DIAGRAM AND SCENARIOS 3](#_Toc498596717)

[WIREFRAMES 4](#_Toc498596718)

[PERSISTENT DATA 4](#_Toc498596719)

[APPENDICES 5](#_Toc498596720)

[APPENDICE A 5](#_Toc498596721)

# PROJECT MANAGEMENT

* We have 3 people working on this project:
  + Duncan Sterken: Backend.
  + Peter Janssen: Front End + Backend
  + Guylian Gilsing: Backend + Frontend.
* Each member will do these things:
  + Duncan Sterken:
    - Backend Chat System.
  + Peter Janssen:
    - Front End design.
    - Backend Login System.
    - Backend Register System.
    - Backend Account Systeem.
  + Guylian Gilsing:
    - Front end Realization.
    - Backend File Uploader.
    - Database Design.
* We use the SCRUM principle, this means that we hold meetings at the start of the day.
* Every morning.

# PROJECT DETAILS

* Project Name: The I.T. Connection.
* Name of the client: P. Nocker.
* Name of the project Guylian Gilsing’;

# PROJECT DESCRIPTION

* At this moment there isn’t a hub for people in the I.T. work field. Our goal is to make a website that provides a platform for these people to network and help each other.

# REQUIREMENTS

Our website needs to do these things:

* Must
  + Contacts (Friend System)
  + Chat System
  + Register System
  + Self-provided user information
  + Forum
  + Coding Languages: HTML – CSS – Javascript – PHP – SQL
  + Easy GUI
  + Framework: Bootstrap
  + Moderators
* Should
  + Easy Contact Information Exchange
  + Fully responsive website
* Could
  + Business Job Offers
  + Business Accounts
  + Electron App (If we have a lot of time left)
* Won’t
  + Existing Social Media Clone
  + Complex GUI

Requirements can be divided in the following categories:

* Must
* Should
* Could
* Won’t

We called this: MoSCoW

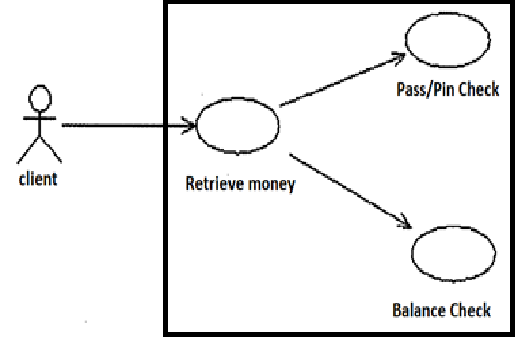
Our client (P. Nocker) expects a working social media website that meets all the requirements mentioned above. We (Waifu Logics) are doing our best to realize this project.

# USE CASE DIAGRAM AND SCENARIOS

Based on the required functionality of the system, we created some use case diagrams:

How to make them:

1. Which ACTOR(s) will work with the system?
2. What functionality will an ACTOR use/have available.
3. Each functionality translates to a use case.
4. When a (sub)functionality is used by multiple actors you must make it a separate use case.
5. Each use case diagram has the following general outline



1. For each use case you should write a scenario. (What has to be done to use the functionality in steps).
2. The general outline of a scenario:

Name: << name of the use case >>

Actor : << name of the actor(s) who use this >>

Pre : << conditions which should be satisfied before the use case can be used >>

Description: << the normal flow of the use case in steps (functional) >>

Alternative: << the alternative flow(s) if possible (e.g.: error flow)

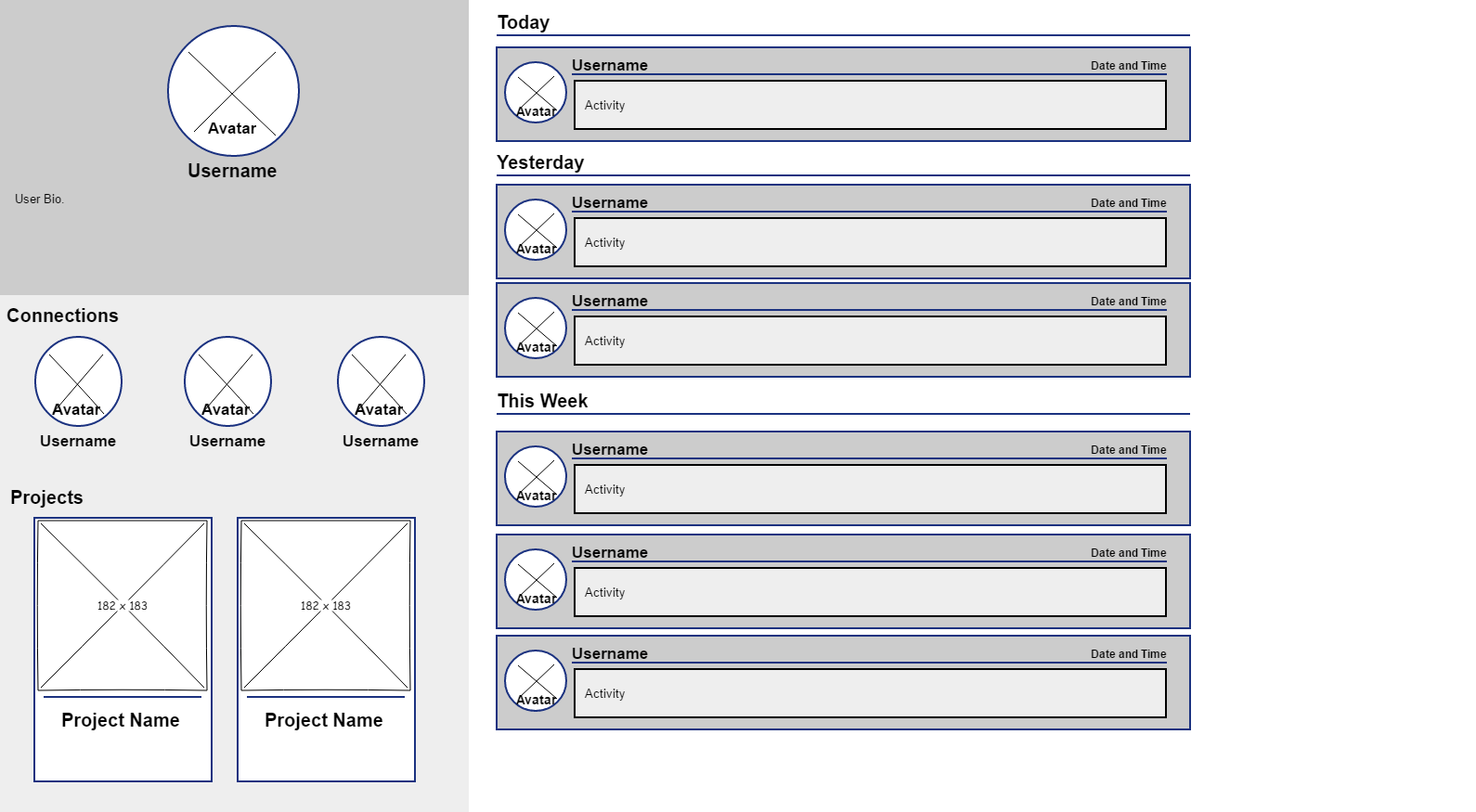
Result: << the expected result of the normal flow>>

You should check if the use cases cover ALL the requirements ( use a matrix to show that)

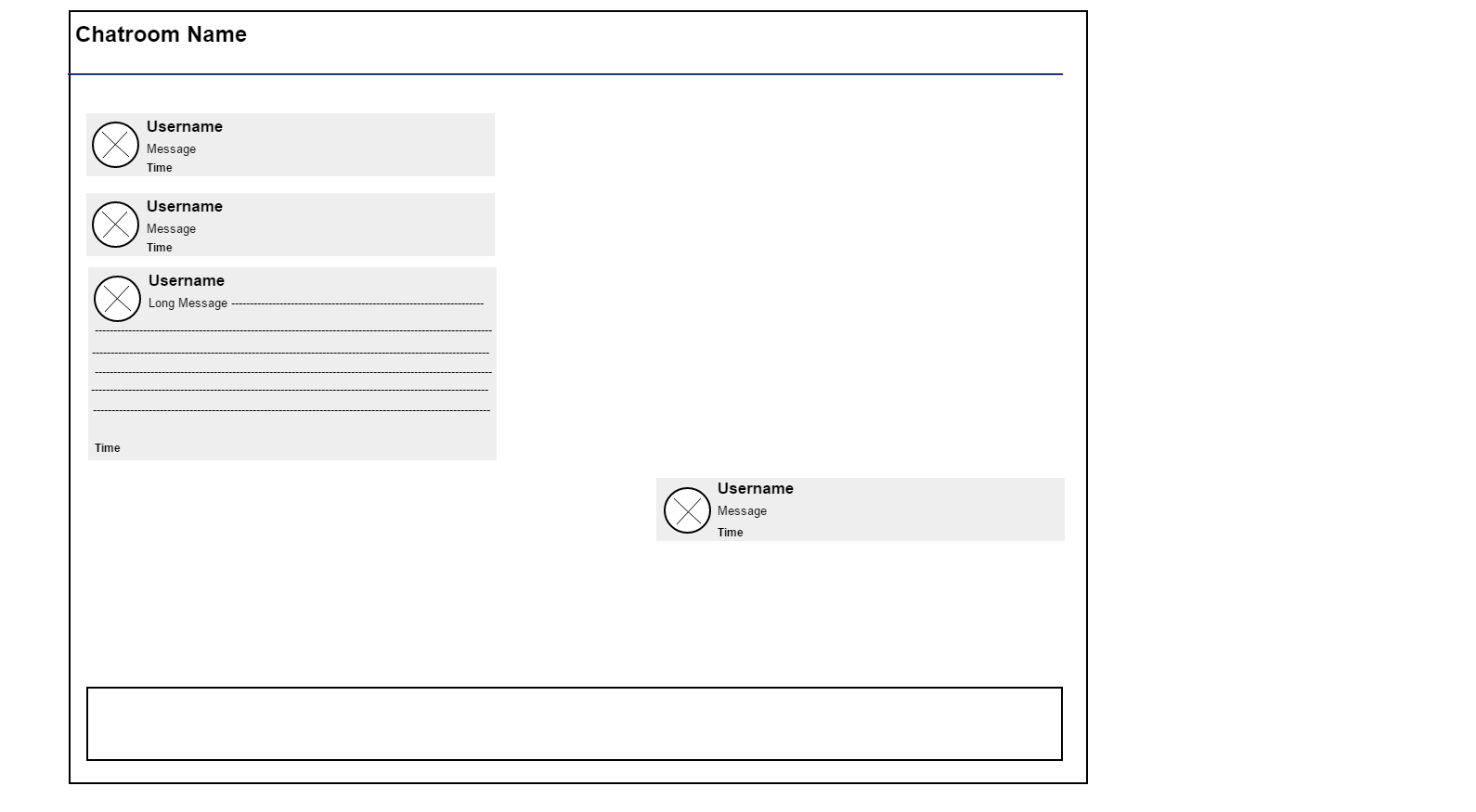
# WIREFRAMES

We also created some wireframes that will give the client some vision in how the website is going to look.

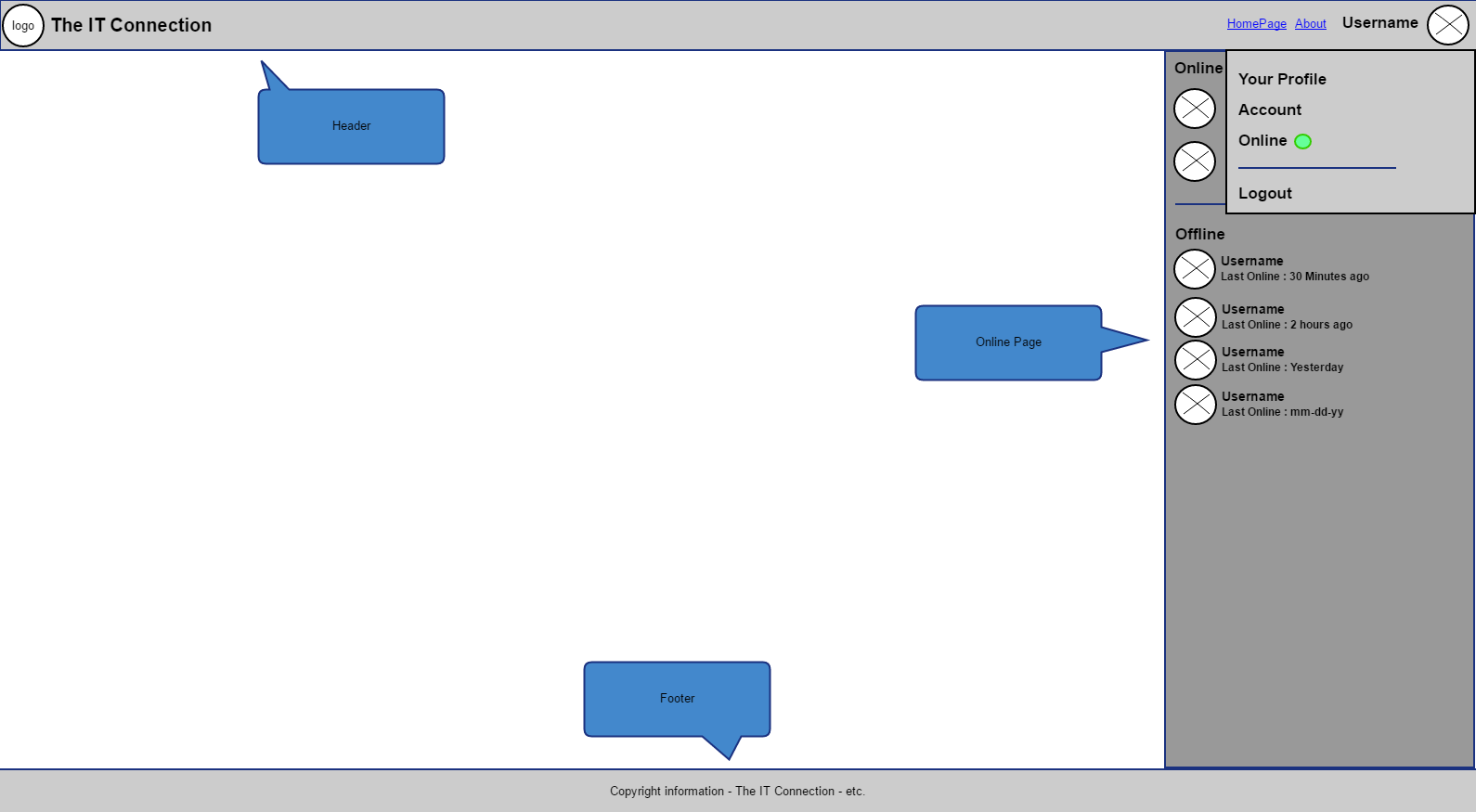
User account page:



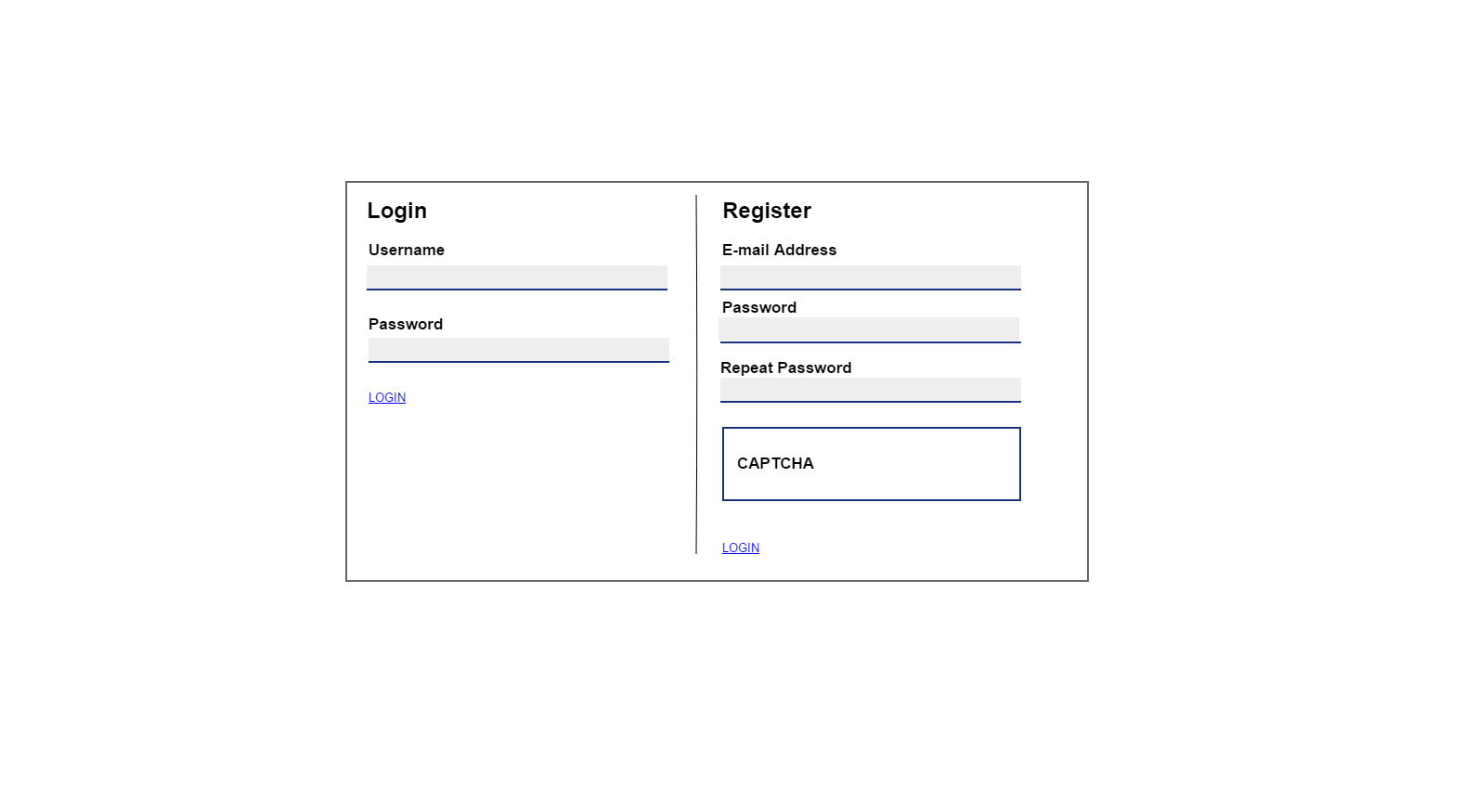
Chat room:



Home page:



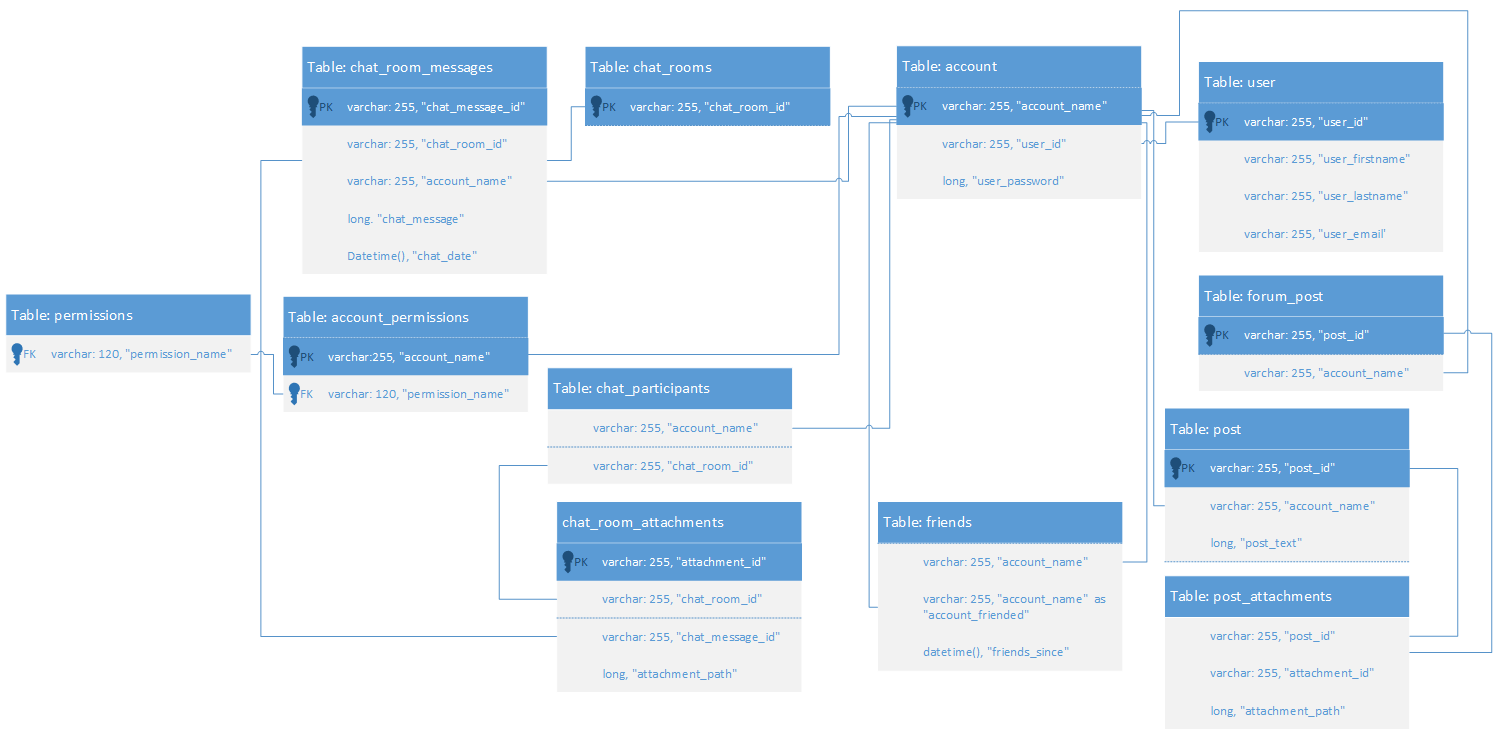
Register Page:



You should check if the wire frames cover ALL the use cases ( use a matrix to show that)

# PERSISTENT DATA

We have a lot of data we need to handle, to visualize the database that needs to store this data we created an ERD.

APPENDICES

## APPENDICE A

Approval.

If You agree with the content of this Functional Design Document, please return a signed copy of it.