**Software Requirements and Design Document**

**For**

**Group 27**

Version 1.0

**Authors**:

Ian Estevez

Jack Throdahl

William Hudmon

# Overview (5 points)

GrouPay is a web application designed to facilitate users in reconciling collective debts, by allowing individuals to join billing groups, in which they can independently contribute to bills owed by themselves and their peers within the same billing group. Users must first make accounts to find the other users with which to join the same billing group, set up by a third party such as a landlord (or property manager, company, etc.), after which they can split bills, such as rent or utilities evenly, and/or distribute percentages of the balance due however they decide. This is done using a MySQL database and using Python with Flask to connect to a web app and read the database.

Functional Requirements (10 points)

1. Users should be able to register to an account and create a unique username, enter their first and last name and a password. This is of high priority
2. Using their username and password they can log into an account and search for users, add them, and invite them to a billing group. This is of high priority
3. Users can join billing groups when invited or create one, and make collective payments to a bill. This is of high priority
4. Users can also communicate with the creator of the group to change the distribution of the balance. This is of high priority.
5. Companies should be able to make an account similarly with company name and be able to make multiple billing groups. This is of medium priority
6. Companies should be able to delegate percentages of shared expenses of each billing group to the users of that group. This is of medium priority
7. Companies can add or remove any user from a billing group they created. This is of medium priority.

# Non-functional Requirements (10 points)

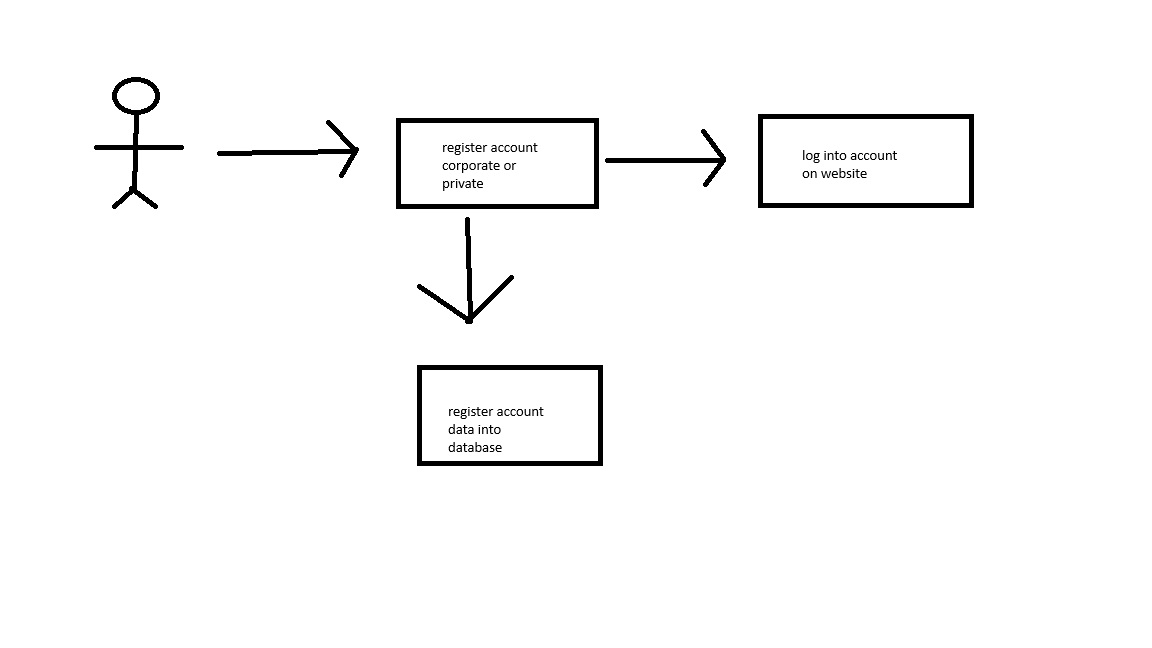
* The database should be distributed and be able to be accessed from anywhere.
* The site should be secure such that no user can read data from another user.

# Use Case Diagram (10 points)

*A diagram of a person's hand

Description automatically generated*

# Class Diagram and/or Sequence Diagrams (15 points)

**