**Progress Report**

**- Increment 1 -**

**Group #27**

*Please use this template to describe your progress on the group project in the latest increment. Please do not change the font, font size, margins or line spacing. All the text in italic should be removed from your final submission.*

# Team Members

* Ian Estevez – iae21 – IAE21
* William Hudmon – wsh19a – Hudbone
* Jack Throdahl – jtt20q – throdahl

1. **Project Title and Description**

This project is titled GrouPay. 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 in order 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 way, users are individually held accountable for their own parts of the total balance due, simplifying the process of collectively managing mutual expenses.

1. **Accomplishments and overall project status during this increment**

During this increment, we’ve accomplished setting up the MySQL database, begun the implementation of the two most relevant tables representing the two main features of the application (users and billing groups), set up the basis for the user experience, and worked over some minor error checking (duplicate user prevention, input validation, etc.). Moreover, the initial Python modules and HTML pages allowing for rudimentary site navigation and basic business logic have been created with Flask, as well as user persistence down to the database level have been implemented, such that one is now able to launch the web app, register for both private and corporate accounts, log in to the GrouPay system, and reach a home page/dashboard. The registration system successfully adds a new user to the MySQL database, providing persistence for that user (although it is currently all stored locally) and ensuring that all records therein are unique, such that no one user can register for more than one account of the same type. Additionally, the login page does input validation, checking the submitted credentials against the database to ensure the user exists before allowing them to log in, and will flash a message indicating incorrect credentials when applicable. Currently, the home page sits empty, as billing groups have not yet been implemented outside of the database, and the register page may require some amount of JavaScript to accomplish adding the “Company Name” field dynamically, but a very bare bones skeleton of our original proposal has been completed as of 10/11/2024.

1. **Challenges, changes in the plan and scope of the project and things that went wrong during this increment**

The most challenging aspect of this increment was reconciling issues caused by the different operating systems used during the development of the code we’ve implemented so far. For William, using a Python venv for development and PyCharm as his IDE of choice, there were no issues relating to accessing the MySQL server locally on his Windows machine. However, for Ian and Jack, running the project using Ubuntu, and its default out-of-the-box installations of MySQL, the default authentication plugin for the user ‘root’ needed to be changed, and a password needed to be set, as the latest Ubuntu version of MySQL had a deprecated authentication plugin, which also did not require a password. This caused not only merge conflicts within the repo but errors for Ian and Jack when running any version of the project that William had worked on, considering his commits involved specifying a password for the user ‘root’@’localhost’ because his operating system required it from the beginning, leading to many hours of foraging through Stack-overflow for solutions as to how to get the project running for all those involved. However, this was eventually solved, and instructions and documentation for those running the project in Ubuntu have been specified within the README.md file.

As for changes to the initial plan and overall scope, our idea has not changed, but we are quickly realizing that running the entire project on ‘localhost’ may not be entirely viable, and that changes may need to be made to the database schema involving the persistence of users, billing groups, and their relationships, considering that multiple users can be in multiple billing groups, and this needs to be reflected and recorded within the database.

1. **Team Member Contribution for this increment**

* Ian Estevez:
  + Progress Report: Primarily wrote all sections of the progress report except for Stakeholder Communication, with input/suggestions from and discussion with William.
  + Requirements and Design Document: Gave input/suggestions and discussion for the sections of the document relevant to Increment 1.
  + Implementation and Testing Document: Primarily wrote the two sections of the document required for Increment 1, with input/suggestions from and discussion with William.
  + Source Code: Finalized the primary business logic Python Flask module, allowing users to navigate the site along with error checking on the forms and within the database (preventing duplicate users and incorrect credentials for logging in); Fully realized the creation of the account registration page, allowing users to choose between corporate and private accounts; Tweaked presentation of all current HTML pages to center page content across the web application; Tweaked database schema to include fields earlier discussed within project proposal for both users and billing groups; Merged several setup files into a single script that is run automatically upon launching the project; Wrote all documentation included in README and instructions for Ubuntu users.
  + Video: Appears in and contributes to discussion within video; filmed, saved, and submitted video.
* William Hudmon:
  + Progress Report: Gave input/suggestions and discussion for the sections of the document relevant to Increment 1.
  + Requirements and Design Document: Primarily wrote all sections of the Requirements and Design Document, with input/suggestions from and discussion with Ian.
  + Implementation and Testing Document: Gave input/suggestions and discussion for the two sections of the document required for Increment 1.
  + Source Code: Created the MySQL database, and the tables representing the application’s users and billing groups; Added code fixing issues connecting to MySQL database; Created admin user with all privileges within database; Set password for both root and admin external database users.
  + Video: Appears in and contributes to discussion within video.
* Jack Throdahl:
  + Progress Report: Primarily wrote Stakeholder Communication for the progress report.
  + Requirements and Design Document: Absent for the drafting of this document; no contribution.
  + Implementation and Testing Document: Absent for the drafting of this document; no contribution.
  + Source Code: Created the primary Flask module responsible for routing between pages, connecting to the MySQL database, and inserting newly created accounts down to the database level; created the first, most relevant HTML pages of the site (login.html, register.html, and home.html) allowing a user to register for an account, log into the account, and reach a home page.
  + Video: Absent for the recording of the video; no contribution.

1. **Plans for the next increment**

By Increment 2, we plan to have a more developed, populated front-end and user experience rather than plain, whitespace HTML forms being presented to the user, including a fully realized dashboard/landing page for authorized, logged in users of the application. This page will hold all billing groups a user is part of, and will feature allowing users to search for, join, and send invites to billing groups. Moreover, we plan to begin implementation of billing groups within the system, including creating billing groups (only accessible to corporate users), joining billing groups, setting bill amounts, and adding persistence to billing groups and their constituents, such that there are records of which users are part of which billing groups, and the respective percentages of the shared expense that each user is obligated to pay. Otherwise, we plan to begin distributing the MySQL database back-end, such that it no longer operates locally on the client machine, and instead can be accessed from anywhere, meaning the data is stored server-side, while being read from and interacted with by individual clients through their web browsers.

1. **Stakeholder Communication**

Dear Stakeholders,

I am pleased to share the progress we have made on the development of GrouPay. Our team has been working hard to build a platform that enhances the group payment user experience. So far, we have successfully implemented the foundational database infrastructure for GrouPay. This includes establishing tables for users and billing groups. The USERS table stores user information while the BILLGROUPS table manages the group payment details. The application has support for user registration and login. New users can create an account by providing their first and last name, username, and password. This registration data is then stored in the USERS table. In the future we will fully implement login functionality along with a personalized user experience. After this we will begin implementing more functionality into GrouPay. This includes creating new groups, inviting other users to groups, funding group wallets, and managing shared bills. We will implement thorough testing throughout the development process to ensure the development of a good working product. Feel free to reach out if you have any questions or would like more detailed information about the GrouPay development process.

Regards,

GrouPay Development Team

1. **Link to video**

<https://youtu.be/AbRB2GASbv0>