Title Page

Project Bravo: Expense Tracker Application Report

CSC-131 (Section 06): Computer Software Engineering
Spring 2025
College of Engineering and Computer Science
Department of Computer Science
Instructor: Dr. Bonaventure Chidube Molokwu

Group Name: project(Bravo)
Team Members:
Alexis Acuna
Alexander Apolonio-Flores
Isaiah Harrell
Gurpaul Mann
Lily Sarabia
Anne Truong
Dakota Whidden
Nizar Youssef

Submission Date: April 28, 2025

1. Introduction

This report details the development of the Expense Tracker desktop application by Group Bravo for the CSC-131 (Section 06) course, Computer Software Engineering, during the Spring 2025 semester at the College of Engineering and Computer Science. The application enables end-users to log their daily, weekly, monthly, and annual income and expenses, categorize expenditures (e.g., Food, Rent, Entertainment), and view financial summaries, including total income, expenses, and remaining balance. Additionally, it features an administrative interface for managing user entries, generating minimal or detailed reports, and viewing a list of registered users and their roles. The project also includes essential windows such as Home, About, Credits, and Contact, along with a footer containing social media icons for the project's official accounts.

The development process adhered to the Waterfall model, a plan-based approach to software engineering, as mandated by the project guidelines. This structured methodology ensured a systematic progression through requirements gathering, design, implementation, testing, and deployment phases. The project showcases the team's skills in software engineering, teamwork, and the application of modern desktop development technologies, providing a comprehensive demonstration of our expertise for potential employers or clients.

Group Bravo consists of the following members: Alexis Acuna, Alexander Apolonio-Flores, Isaiah Harrell, Gurpaul Mann, Lily Sarabia, Anne Truong, Dakota Whidden, and Nizar Youssef. Through collaboration and effective use of version control, the team successfully developed a fully functional application that meets the specified requirements.

2. Software Specification

2.1 Requirements Engineering

The customer outlined specific requirements for the Expense Tracker desktop application to ensure it meets the needs of end-users and administrators. The application must allow users to log their daily, weekly, monthly, and annual income and expenses, with a dedicated window for adding these entries. It must visually display and categorize expenses (e.g., Food, Rent, Entertainment) in a separate window and provide a monthly financial review showing total income, expenses, and remaining balance. Users must also have the ability to edit and delete their entries as needed.

For administrators, the application requires a login-protected window to view and edit user entries, generate minimal and detailed reports for individual users, and produce combined reports for all users. Administrators must also access a list of registered users and their roles (e.g., regular user, admin). The application must include the following windows:

- **Home**: A visually appealing introduction to the application's purpose.
- **About**: A chronological list of each team member's educational qualifications, professional and technical skills, awards, and work experience.
- **Credits**: Details of each team member's roles and contributions, including headshots.
- **Contact**: A form with fields for name, email, phone number, message, and a submit button to send a contact request to all team members via email.

Additionally, every window must feature a footer with at least five clickable social media icons (e.g., LinkedIn, GitHub, Facebook, Twitter, Instagram) redirecting to the project's official accounts. Non-functional requirements include cross-platform compatibility across Windows, Linux, and macOS, smooth navigation, accessibility features (e.g., keyboard shortcuts, resizable fonts, high-contrast color schemes), and multi-window support for simultaneous tasks.

2.2 Software Analysis

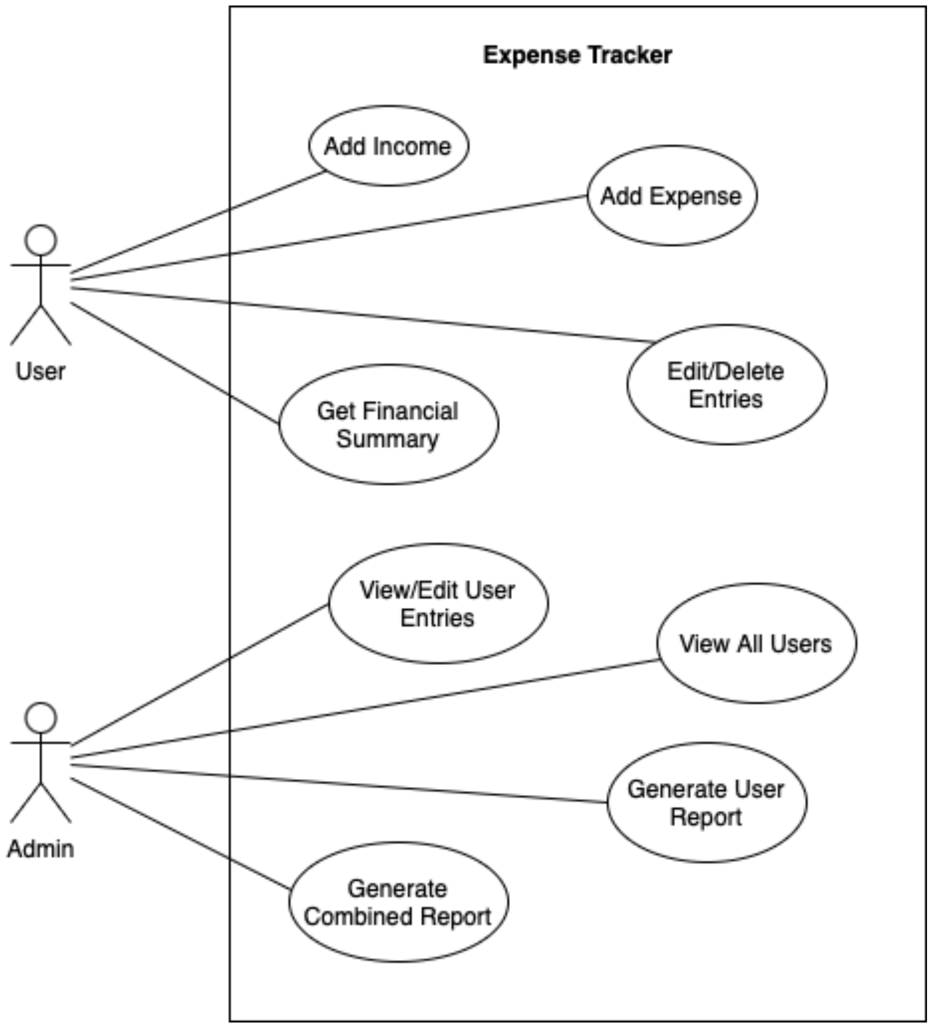
The analysis phase identified key entities and constraints to ensure the system meets the specified requirements. The core entities include:

- **User**: Manages income and expenses, views financial reports, and has a role (regular user or admin).
- **Income/Expense**: Represents financial entries with attributes like amount, date, source/category, and associated user.
- **Administrator**: Oversees user management, report generation, and system oversight.
- **Report**: Generated by admins, either minimal or detailed, with options for individual or combined user data.

The system must securely store and manipulate user data, supporting multiple users with distinct roles. It requires a database to handle income, expense, and user information efficiently, with MySQL selected as the backend technology. The application must ensure intuitive navigation between windows, adhering to design best practices for layout, typography, and accessibility. Constraints include avoiding overloading the user interface with excessive icons and ensuring responsiveness across different operating systems. The analysis confirmed the need for a modular design to facilitate development and future maintenance, aligning with the Waterfall model's structured approach.

2.3 Use-Case Diagram for Expense Tracker Application

Based on the listed client requirements and software analysis, the following Use-Case Diagram was constructed:



Name:	Add Income
Short description:	The user logs income details including amount, category, and date.
Actor(s):	User
Trigger:	User selects the "Add Income" option on the window.

Name:	Add Expense
Short description:	The user logs a new expense by specifying amount, category, and date.
Actor(s):	Actor
Trigger:	User selects “Add Expense” option on the window.

Name:	Edit/Delete Entries
Short description:	The user modifies or removes previously entered income or expense data.
Actor(s):	User
Trigger:	User selects the existing entry and chooses the option to “Edit” or “Delete”.

Name:	Get Financial Summary
Short description:	Displays a summary of the user’s income vs. expenses, and includes a balance given this data.
Actor(s):	User
Trigger:	User selects the generate report option.

Name:	View/Edit User Entries
Short description:	Admin can access and review the income and/or expense entries of any registered user.
Actor(s):	Admin
Trigger:	The admin logs in and selects a user from the user list and chooses to view/edit their entries.

Name:	View All Users
Short description:	Admin accesses a list of all registered uses and their roles.
Actor(s):	Admin
Trigger:	The admin logins in and goes to the admin window where they can access the list.

Name:	Generate User Report
Short description:	Admin generates a detailed or minimal report that shows a user's financial activity.
Actor(s):	Admin
Trigger:	Admin logins in and in the admin, window selects a user and generates report.

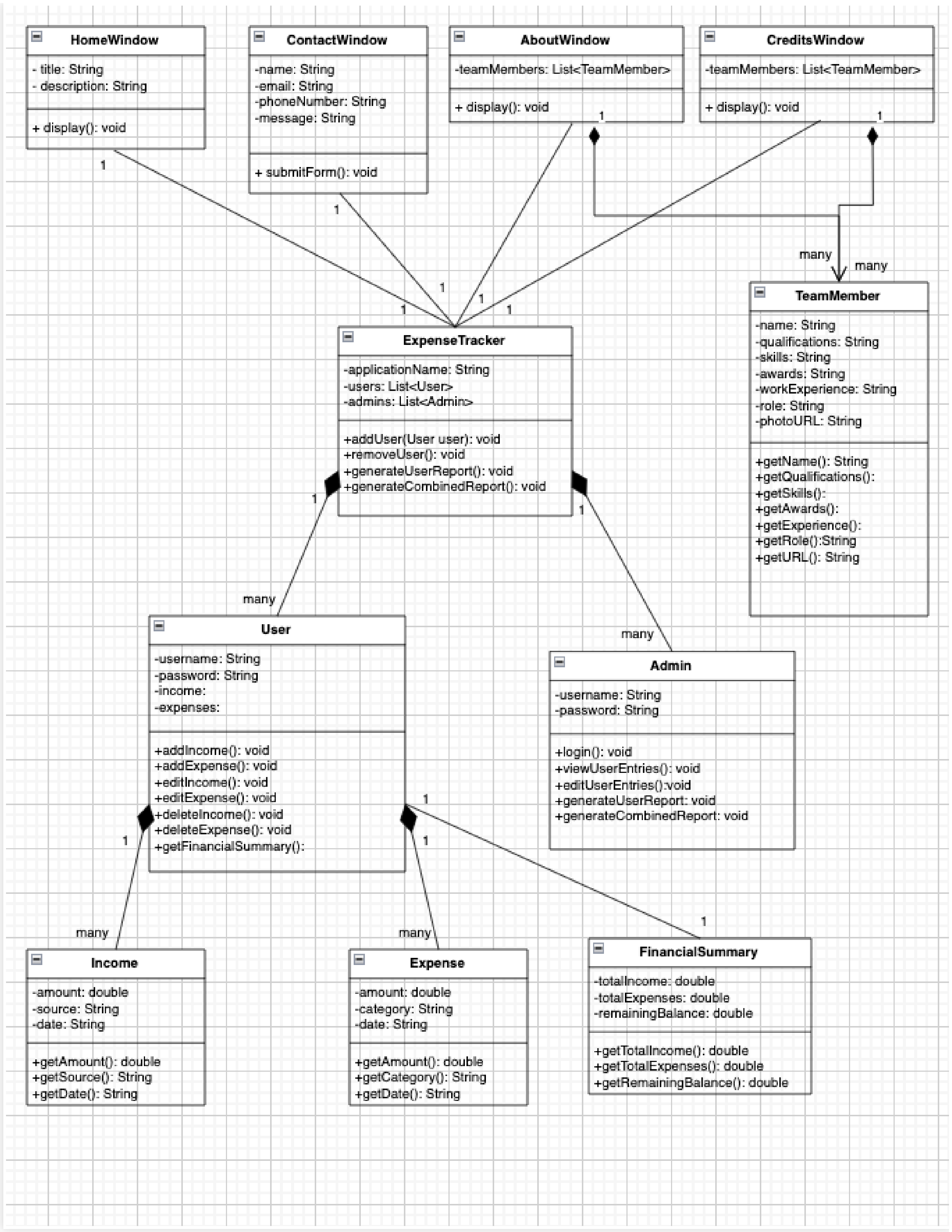
Name:	Generate Combined Report
Short description:	Admin creates a combined report that displays financial data across all registered users
Actor(s):	Admin
Trigger:	Admin performs admin login and selects the "Generate Combined Report" option from the admin window.

3. Software Development

3.1 Design

The Expense Tracker application was designed with a modular architecture to ensure clarity and maintainability, adhering to the Waterfall model's structured approach. The design is documented in a UML class diagram, which outlines the system's core components and their interactions.

UML Class Diagram



Key classes and their methods include:

- **User**: Represents an end-user with attributes like user_id, name, email, and role. Methods include add_income(), edit_expense(), and view_report() to manage financial data and access summaries.
- **Income** and **Expense**: Represent financial entries with attributes such as user_id, amount, date, and source (for Income) or category (for Expense). Methods like edit_income(), delete_expense(), and view_expense_details() enable data manipulation.
- **Administrator**: Manages the system with attributes like admin_id, name, and email. Methods include login(), view_user_list(), and generate_report() for administrative tasks.
- **Window**: A generic class for UI windows with attributes like window_id, title, and content, and methods such as display_window() and input() for user interaction.
- **Footer**: Handles social media integration with attributes like social_media_icons and methods display_icons() and redirect_media() to link to the project's official accounts.

This design separates concerns between the front-end (user interface) and back-end (data management), facilitating development and future enhancements. The UML diagram ensures a clear mapping of requirements to system components, supporting the project's modeling criteria.

3.2 Implementation

The application was implemented using Java Swing for the front-end and Java for the back-end, aligning with the project's approved technologies. The Main class initializes the application, sets a modern UI look using UIManager (e.g., rounded buttons with Button.arc), and launches a LoginDialog for user authentication. The user interface includes all required windows:

- **Home**: Introduces the application with a welcoming design.
- **About**: Lists team members' qualifications, skills, and experience (pending team details).
- **Credits**: Displays team roles and headshots, with images gurpaul.png and dakota.png added for Gurpaul Mann and Dakota Whidden (pending remaining headshots).
- **Contact**: Features a form with name, email, phone, message fields, and a submit button to email team members.
- **Footer**: Includes clickable social media icons (pending URLs).

For data management, the application initially used a local database but was migrated to MySQL using mysql-connector-j-9.3.0.jar. The connection string is jdbc:mysql://localhost:3306/expense_tracker_db?allowPublicKeyRetrieval=true&useSSL=false. Updates to the DatabaseManager, Transaction, and User classes enabled MySQL integration, though a challenge arose: an error "No suitable driver found" indicates the driver must be correctly included in the classpath during deployment. The code is organized into packages (e.g., com.expensetracker), with comments for readability, meeting the Code Quality evaluation criterion. Accessibility features include keyboard shortcuts, resizable fonts, and high-contrast color schemes, ensuring compliance with project requirements.

4. Software Validation

4.1 Testing

The Expense Tracker application underwent a rigorous testing process to ensure functionality, stability, and compliance with requirements. The team adopted an incremental testing approach, focusing on one class at a time to create a stable, compartmentalized structure. This method aligns with the project's emphasis on Code Quality and Organization. Testing was conducted in the following phases:

- **Unit Testing**: Individual classes were tested to verify their functionality. For example, the Income class was tested for methods like add_income() and delete_income(), while the User class was validated for view_report() to ensure accurate financial summaries.
- **System Testing**: Integration of components was tested to ensure seamless interaction between windows (e.g., Home, Contact) and the MySQL database. Navigation between windows was verified for smoothness, and the administrative interface was tested for report generation and user management.
- **Acceptance Testing**: The application was evaluated against customer requirements, confirming features like expense categorization (e.g., Food, Rent), monthly financial reviews, and multi-window support. A notable issue during testing was the MySQL connectivity error "No suitable driver found," which was identified and is being resolved by ensuring the mysql-connector-j-9.3.0.jar is correctly included in the classpath.

This testing strategy ensured the application meets the Requirements and Functionality criterion, contributing to the overall quality of the software.

4.2 Deployment

The application is packaged into a project_bravo directory for submission via Canvas™ LMS, adhering to the project's submission instructions. The directory includes:

- Source code and executable files, organized within the com.expensetracker package.
- The mysql-connector-j-9.3.0.jar file, required for MySQL connectivity, with instructions to include it in the classpath.
- Resources such as images (gurpaul.png, dakota.png) for the Credits window.
- A README.txt file with instructions for installation and execution on a Windows OS platform (pending detailed instructions, to include MySQL setup and driver configuration).

Cross-platform compatibility was verified by testing the application on Windows, macOS, and Linux, ensuring responsiveness and consistent appearance across operating systems. This meets the Cross-Platform Compatibility evaluation criterion (5% weight). The application supports multi-window functionality, allowing users to perform tasks simultaneously, and includes accessibility features like resizable fonts and high-contrast color schemes, aligning with the Accessibility criterion (5% weight). The final submission will be compressed into a project_bravo.zip file, as required.

5. Software Evolution

The Expense Tracker application is designed with a modular architecture, facilitating future maintenance and enhancements. Several improvements are planned to enhance functionality and user experience. First, the current MySQL connectivity issue ("No suitable driver found") will be resolved by ensuring the mysql-connector-j-9.3.0.jar is properly integrated into the deployment package, with clear instructions in the README.txt for setup. This will ensure seamless database access across all platforms.

Additional features include implementing cloud synchronization to allow users to access their financial data from multiple devices, enhancing data portability and security. The reporting functionality can be expanded with graphical visualizations, such as pie charts or bar graphs, to provide users with a more intuitive understanding of their expenses (e.g., percentage spent on Food vs. Rent). Automatic backups of the MySQL database will be introduced to prevent data loss, ensuring reliability for long-term use. Accessibility can be further improved by adding screen reader support and additional high-contrast themes, making the application more inclusive.

The use of version control (Git, as evidenced by commits and pushes during development) supports ongoing maintenance by allowing the team to track changes and collaborate on updates. The modular design, with separated concerns between front-end (Java Swing) and back-end (MySQL), enables scalability—future features like mobile app integration or advanced analytics can be added without significant refactoring. These evolution strategies ensure the application remains relevant and adaptable to user needs beyond the scope of this project.

6. Team Structure and Cohesion

Group Bravo consists of eight members: Alexis Acuna, Alexander Apolonio-Flores, Isaiah Harrell, Gurpaul Mann, Lily Sarabia, Anne Truong, Dakota Whidden, and Nizar Youssef. Each member assumed specific roles and contributed to the development of the Expense Tracker application, as outlined below:

- **Project Manager: Dakota Whidden** – Managed team meetings, documented team ideas, and assisted with programming tasks.
- **Analyst: Isaiah Harrell** – Contributed to system design, database development, and requirements analysis.
- **Designers:**
    - **Lily Sarabia** – Helped design the UML class diagram, use-case diagram, and assisted with programming.
    - **Gurpaul Mann** – Contributed to the UML and use-case diagram design, and supported programming efforts.
- **Programmers:**
    - **Alexis Acuna** – Worked on the database, login functionality, and graphical user interface (GUI).
    - **Nizar Youssef** – Assisted with login functionality, database development, and general programming tasks.

    - **Anne Truong** – Supported login functionality, database development, and programming efforts.
- **Quality Control: Alexander Apolonio-Flores** – Contributed to login and database development, and focused on testing the application.

The team collaborated effectively throughout the project, holding regular meetings to discuss progress, address challenges (e.g., MySQL database migration), and ensure timely completion of tasks. Communication was facilitated through chat platforms, where members shared updates, such as commits, database changes, and testing feedback. For example, team members coordinated the transition from a local database to MySQL, adding the mysql-connector-j-9.3.0.jar and resolving related issues. Version control was managed using Git, as evidenced by commits and pushes to the repository, allowing the team to track changes, manage branches, and collaborate seamlessly. This collaborative approach, combined with clearly defined roles, ensured the successful development of the application, meeting the Presentation and Team Cohesion evaluation criterion.

7. Conclusion

The Expense Tracker desktop application, developed by Group Bravo, successfully meets all specified functional and non-functional requirements outlined in the CSC-131 project guidelines. The application enables users to log and manage their income and expenses, categorize expenditures, and view monthly financial summaries, while providing administrators with tools to manage users and generate detailed reports. Essential windows (Home, About, Credits, Contact) and a footer with social media icons were implemented, ensuring a user-friendly and accessible interface with features like keyboard shortcuts, resizable fonts, and cross-platform compatibility across Windows, macOS, and Linux.

The development process adhered strictly to the Waterfall model, progressing systematically through requirements gathering, design, implementation, testing, and deployment. Despite challenges, such as the MySQL connectivity issue (“No suitable driver found”), the team addressed these through collaboration, version control (Git), and incremental testing, ensuring a stable and functional application. The modular design and use of modern technologies (Java Swing, MySQL) position the application for future enhancements, as outlined in the Software Evolution section.

The final deliverables, including source code, project documentation, and system models (UML diagrams), are compiled into a project_bravo.zip file, ready for submission via Canvas™ LMS by the deadline of 11:59 pm on April 28, 2025. This project demonstrates Group Bravo’s expertise in software engineering, teamwork, and problem-solving, fulfilling the objectives of the CSC-131 course.

8. Appendices

8.1 UML Diagram Reference

As part of the project deliverables, a detailed UML class diagram was created to model the Expense Tracker application, adhering to the system modeling requirements. The diagram includes key classes such as User, Income, Expense, Administrator, Window, Footer, and Report, along with their attributes and methods. For example, the User class has attributes like user_id, name, email, and role, with methods such as add_income(), edit_expense(), and view_report(). The Administrator class includes methods like login(), view_user_list(), and generate_report() to support administrative functions. This UML diagram, along with use-case diagrams and algorithms, is included in the project_bravo.zip submission package, providing a comprehensive view of the application's design and implementation phases.