Sweet Ledger

Guideline on Preparing the Project Proposal

The written project proposal you will soon submit should at the minimum have the following sections. Each team will be given 15 minutes to present and discuss their project proposal. Prepare power point slides to guide your oral presentation.

1.1 Title

Sweet Ledger Financial Ledger

1.2 Motivation

Working in Ecommerce will help us expand our portfolio when interviewing for jobs. It helps us expand our knowledge in programming since we are learning languages like react and javascript.

1.3 Statement of the Problem or Need

The project is trying to accomplish a way for users to manage their financial statements in an app and show their spending habits. Allowing the managers and Administrator to properly govern the statements and make sure the accounts are active and accurate.

1.4 Objectives

- Make the website user-friendly
- Make sure the website has dependable software so the journals, trial balance, etc are as accurate as possible.
- Make sure that website has a unified design to it
- Make sure the users are able to transition between the pages as smoothly as possible
- Make sure the security is at the level of the requirements

1.5 Methodology

The project will be divided amongst the group, however if some of the tasks are easier some of the more experienced group members can take on more tasks and more people can work on the tasks that are more in depth. We used Trello to keep track of the requirements of the sprints. The proposal and the trello will be prepared to discuss the program. The group meets after class in the library to discuss topics on how to approach the project and what to work on for about 30 minutes to an hour. We plan to complete the architecture of the website so we can use the pages to work on the different functionalities for the admin, manager and regular users. Also, have the login, logout buttons, back buttons, and clear button fully functional by the end of the first sprint. We are using react, javascript, and mongodbs as the software for the app.

1.6 Knowledge Areas Needed for the Project

The group members will need to have or develop the knowledge to be able to work in visual studio code, code in react and javascript, have some knowledge of the mongo database, and be able understand the accounting domain that was taught in class.

1.7 Project Deliverables and Beneficiaries

Deliverables:

- Interface for entering, categorizing, and managing company expenditures.
- Record revenue streams (sales, services, etc.).
- Trial Balance (debits/credits listing).
- Profit & Loss Statement / Net Income Calculation (revenues – expenses).
- Balance Sheet
- Multiple users with different roles (employee, manager, accountant).
- Access control and permissions.

Beneficiaries

- Business Owners / Entrepreneurs
- Accountants / Finance Teams
- Managers / Department Heads
- External Stakeholders (Investors, Auditors, Banks)

1.8 History

- Similarities:
  - Being as reliable as existing accounting or expense tracker apps (e.g., QuickBooks, Zoho, Wave), our project helps track income and expenses.
  - It can generate financial statements like a trial balance or journals.

- Differencies:
  - It could emphasize simplicity and user-friendliness instead of complicated accounting jargon.
  - We could focus on small businesses or startups who don't need overly complex accounting software.
- Unique:
  - Providing an all-in-one user-friendly tool that combines expense tracking + automated financial statements without requiring advanced accounting knowledge.

1.9 Proprietary Information and Confidentiality Requirements

Proprietary Information:

- Details of income, expenses, assets, liabilities.
- Login credentials, employee expense submissions, and personal identifying information.
- Trial balance, net income, Journals, chart of accounts, etc.

Confidentiality Requirement

- All financial and user data must be stored securely (e.g., encrypted in databases).
- Only authorized users (e.g., business owner, finance manager) should be able to view/edit sensitive financial reports.
- Role-based permissions (e.g., employees can submit expenses but not see full statements).
- Protecting personal data of employees and company stakeholders.
- Ensuring no unauthorized sharing of financial information with third parties.
- Generated reports (trial balance, net income) are for internal use only and must not be exposed publicly.

1.10 Required Facilities

What facilities are needed by team members to work on this project?

- Computers or laptops that can run visual studio code
- A directory to secure the folder and files for the project
- A smart room for our team to provide a space for our team to discuss topics and allows us to help each other if needed.

1.11 Project Assumptions and Constraints

Building the infrastructure like the login screen, pages for the roles and U.I. will be the longest in the planning stage whether it's to determine how the login screen works or how the logo should look or what code languages and database to use. Being able to apply the accounting domain to the software could be another long process.Having minimal or no knowledge of the coding language and not understanding the accounting domain can limit the completion of the project.Also, figuring out how to connect the database to our project limited us at first but we were able to accomplish our goal.

1.12 Major Stakeholders

- The development team
- The customers/ users
- The client

1.13 Class Diagram

```
+-------------------------+
|     FinancialReport     |
+-------------------------+
| - year: int             |
+-------------------------+
| + generateTrialBalance()     |
| + generateIncomeStatement()  |
| + generateBalanceSheet()     |
| + generateCashFlow()         |
| + generateRatioAnalysis()    |
+-------------------------+
```

```
+-------------------------+
|   Document              |
+-------------------------+
| - docId: int            |
| - fileName: String      |
| - fileType: String      |
| - filePath: String      |
+-------------------------+
| + upload()              |
| + download()            |
+-------------------------+
```

```
+--------------------------+
|     Account              |
+--------------------------+
| - accountId: int         |
| - name: String           |
| - number: String         |
| - type: enum {Asset,...} |
| - normalSide: enum {Dr,Cr} |
| - isActive: boolean      |
+--------------------------+
| + deactivate()           |
| + generateReport()       |
+--------------------------+
```

```
+--------------------------+
|     ChartOfAccounts      |
+--------------------------+
| - accounts: List<Account> |
+--------------------------+
| + addAccount(Account)    |
| + findAccount(number)    |
| + listAccounts()         |
+--------------------------+
```

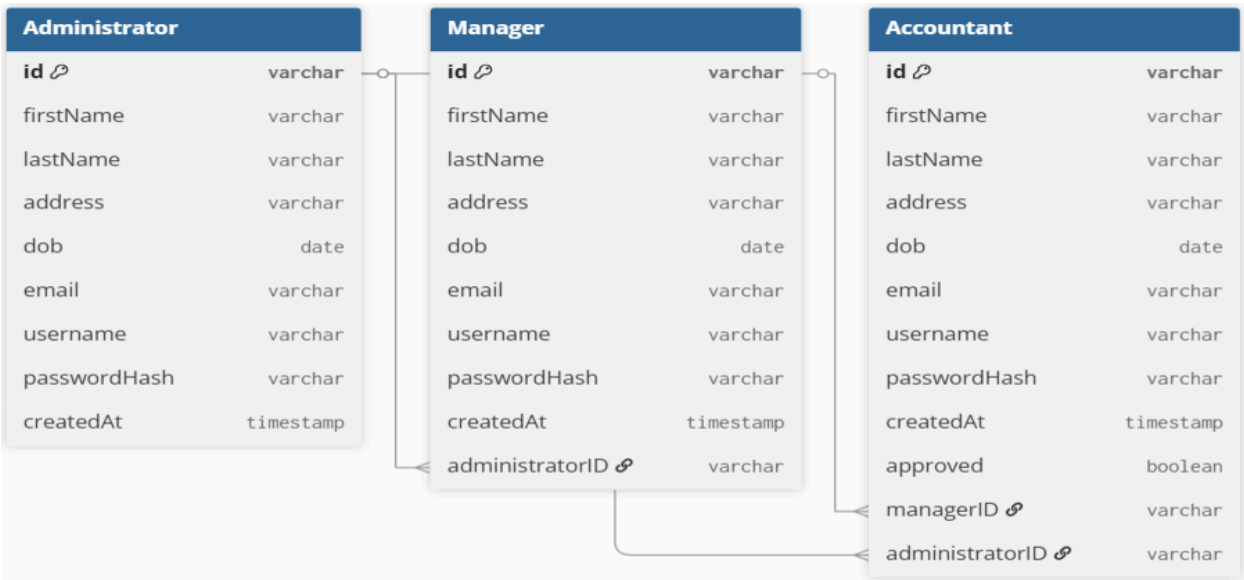
```
+--------------------+     +------------------+
|       User         |<>------|      Role        |
+--------------------+     +------------------+
| - userId: int      |     | - roleId: int    |
| - username: String |     | - roleName: String |
| - passwordHash: Str |    |                  |
| - email: String    |     |                  |
| - isActive: boolean |    +------------------+
+--------------------+
| + login()          |
| + logout()         |
| + resetPassword()  |
| + changePassword() |
+--------------------+
```

```
+----------------------+
|   PasswordPolicy     |
+----------------------+
| - minLength: int     |
| - complexityRules: Str |
| - expirationDays: int |
| - historyCount: int  |
+----------------------+
| + validate(password):bool|
+----------------------+
```

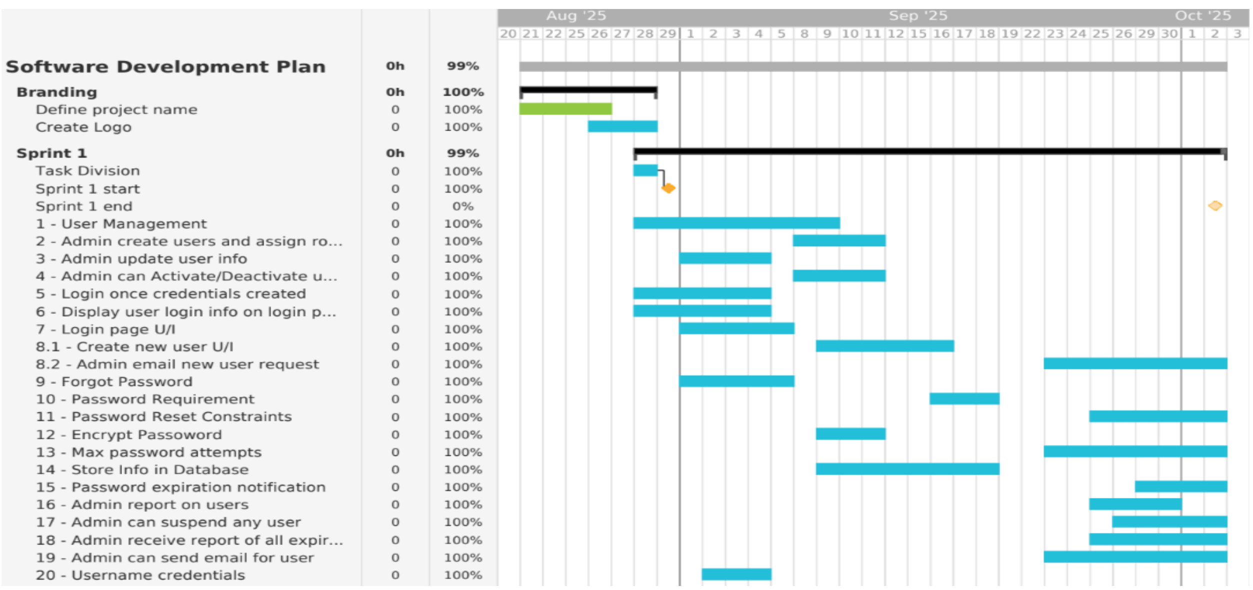
```
+--------------------------+
|   JournalEntry           |
+--------------------------+
| - entryId: int           |
| - date: Date             |
| - status: enum {Draft,Approved,Rejected} |
| - sourceDocs: List<Document> |
+--------------------------+
| + addDebit(Account, amount)  |
| + addCredit(Account, amount) |
| + validateBalance()          |
| + attachDocument(Document)   |
+--------------------------+
```

```
+--------------------------+
|   TransactionLine        |
+--------------------------+
| - lineId: int            |
| - account: Account       |
| - amount: double         |
| - type: enum {Debit,Credit} |
+--------------------------+
```

1.14 Database Diagram



1.15 Project schedule



1.16 Primary Contact

Devin Perry (Team Leader)

Email:dperry52@students.kennesaw.edu

Here are links with reference material:

http://www.ece.rutgers.edu/~marsic/Teaching/SE/proposal.html

http://www.ece.rutgers.edu/~marsic/books/SE/projects/

http://engineering.dartmouth.edu/cook/sample/software/

http://engineering.dartmouth.edu/cook/sample/product-design/

http://engineering.dartmouth.edu/cook/sample/medical/

http://www.ithaca.edu/its/pmo/docs/ProposalExample.pdf