

CMP9134M Advanced Software Engineering

Objective: This project implements the development of the banking system which is designed to handle the sensitive data of the users. This system is built using python and streamlit providing the user friendly web interface for data interaction.

- Agile and XP techniques employed:

Our team adopted agile technology which helped in iterative development of the project.

- User stories: we started with outlining the functionality which are required. So here user stories includes managing user accounts, transactions and providing other services.
- Simple design: We have focused on the development of the system to meet the current requirements without over complicating the architecture.
- Pair programming: Development of this project were often carried out using pair programming which helped to enhance the code quality through the reviews.
- Testing: from the beginning, we implemented the unit testing of all the function to check on reliability.
- Continuous integration and deployment: here in this project, we have used CI/CD pipelines to automate the task of the testing and development.

This structure approached of Agile methodology helped in reliable and smooth development of the project.

- Tools used for the implementation of the project:

- Github : code were managed and version controlled using the github.
- Streamlit: Used for the development of the robust and user friendly web interface.

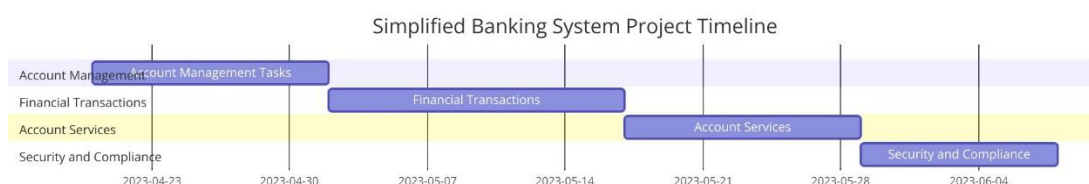
- Roles and responsibilities:

My role can be designated as 'Software developer and tester'. Here is the description of the responsibilities carried out by me during the project.

- User interface development: contributed to the development of streamlit based user interface focusing on the user friendly interface. My aim was to design the webpage to manage the user data and transactions efficiently and securely.
- Unit Testing: Developed unit tests for encrypting the functionalities and user data handling logics using python unittest framework.
- Code review and pair programming: engaged in the pair programming to improve the code quality through the peer reviews. My role was to guide the team members through the complex code sessions.
- Sprint planning: contributed to guide my team in planning the sessions and workflow. Helped the team to prioritize the task and set a goal in project roadmap.
- Documentation: participated in the technical writing of the development and implementation the project.

This efforts were taken to ensure the smooth functionality and secured development of the software.

- Project planning:



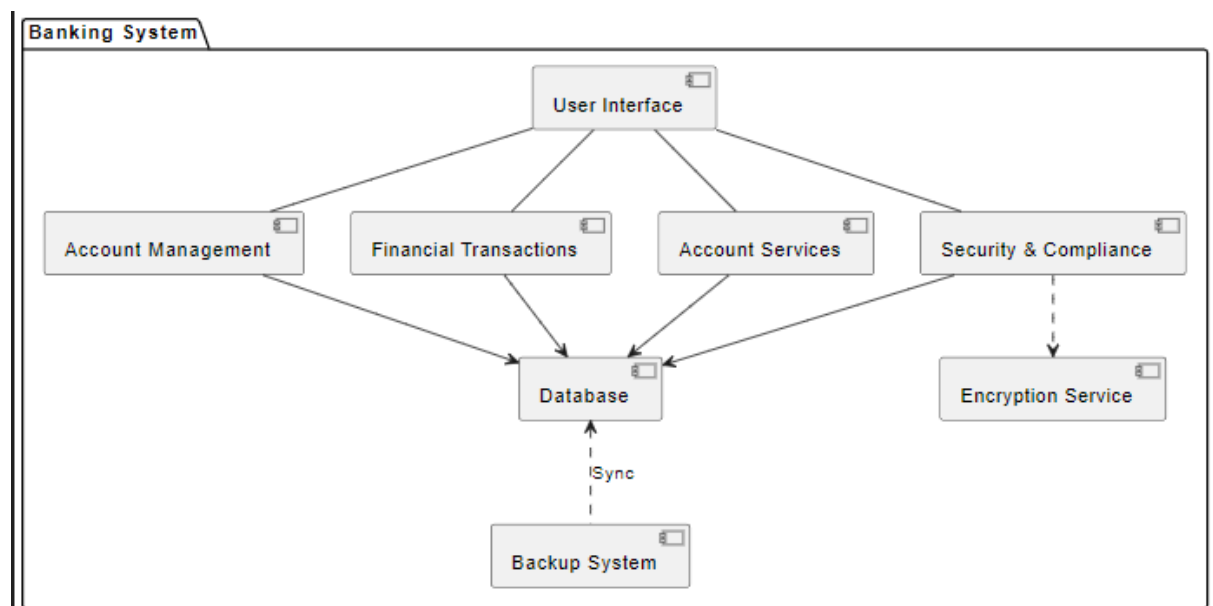
****Gantt chart****

The above diagram represents the project timeline while development of the project. We opted for the agile methodology for the flexibility and the iterative development of the project. We mainly divided this project into four parts i.e. account management, financial transactions management, account services and security and compliance. Thus, we have planned the milestones and completed within the assigned time.

- System design:

The component diagram below represents the banking software working virtually.

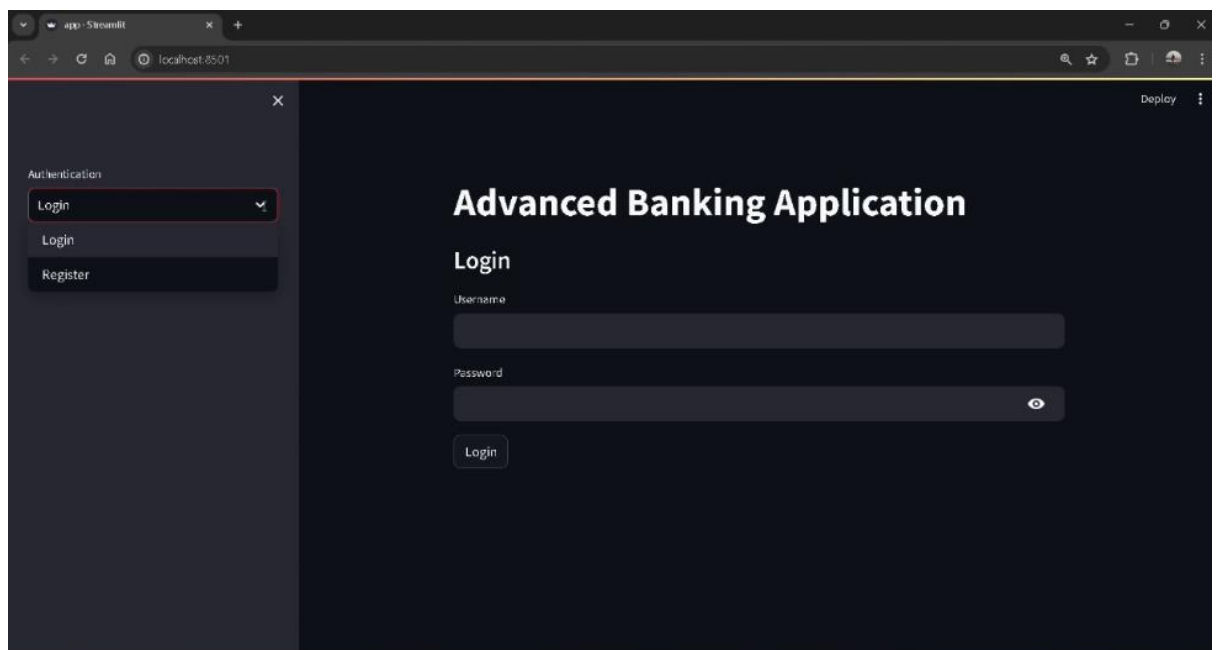
- User interface: it a central point of interaction with the users.
- System: the users are guided to various services such as account management, financial transaction, account services and security and compliances using the user interface.
- Database: the user data generated from the above level is stored in the database.
- Backup system: it ensures the data integrity.
- Encryption services: it is used by security and compliances to secrete the sensitive data of the user such as account number, account password, etc.



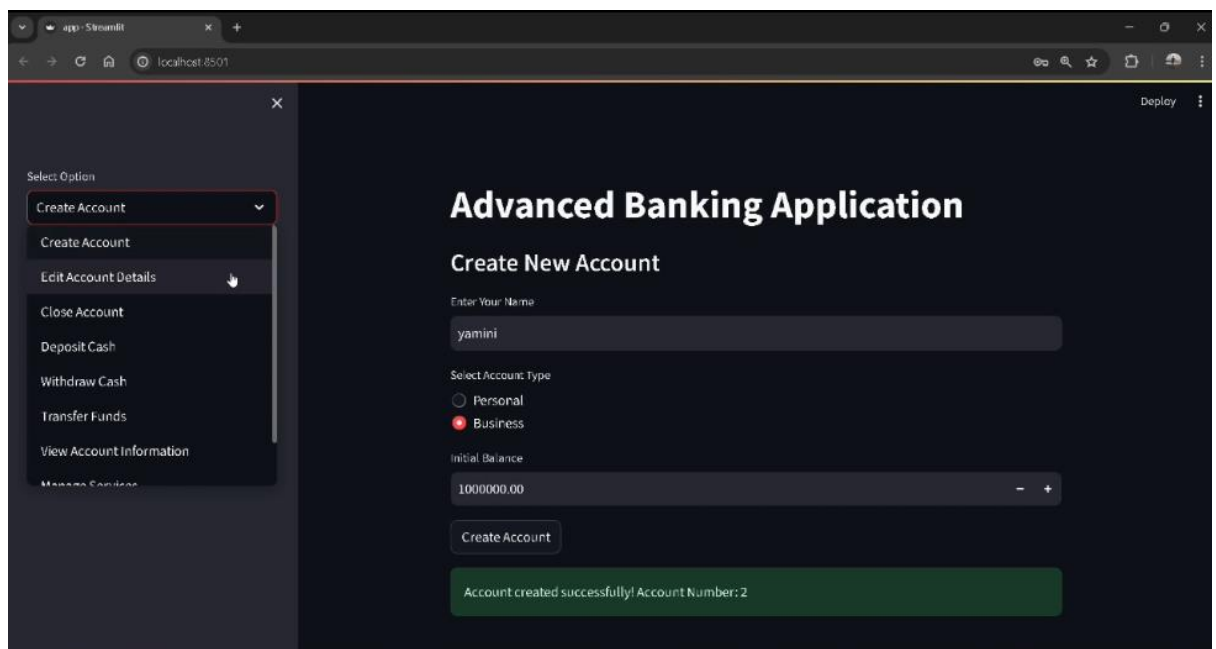
- Graphic user interface design [GUI]:

The GUI is divided into part i.e. main page where user can login using their own credentials or register if they are new and the following page which consist of various functionalities such as creating, editing, deleting the account, withdraw or deposit the cash and many more.

This GUI is designed using the streamlit module which help in designing the robust and user friendly interface without any complication.



****Main page****



****Following page****

Thus, the GUI is so self-explanatory and easy for the users to access their data.

- Version control using git:

We have used github for the version control using the git.

- Code: [**link**](#)
- Documentation [**link**](#)
- Prototypes and testing artefacts: [**link**](#)
- Video :



Banking app demo.mp4

- Pair programming approach:
 - Planning: Before the coding, OOP principles were planned and discussed. The class, methods and hierarchies were decided.
 - Coding: implemented the planned in agile methodology. It was a team approach by switching role constantly to stay actively engaged.
 - Reviewing: discussed and reviewed the code to improve the quality of code as well as fix the bugs.
 - Testing: performed unit testing on the each section of the code for the improvement of the code.
- Implementation:

```
def deposit_cash():
    st.subheader('Deposit Cash')
    account_number = st.number_input('Enter Account Number', format="%.2f")
    amount = st.number_input('Enter Amount', step=0.01, format="%.2f")
    if st.button('Deposit') and account_number and amount > 0:
        if account_number in accounts_df.index:
            accounts_df.loc[account_number, 'Balance'] += amount
            transaction_record = {'Date': datetime.now().isoformat(), 'Type': 'Deposit', 'From': None, 'To': account_number, 'Amount': amount}
            transactions_df.loc[len(transactions_df) + 1] = transaction_record
            save_data(accounts_df, ACCOUNTS_FILE)
            save_data(transactions_df, TRANSACTIONS_FILE)
            st.success('Deposit successful!')
            logging.info(f"Deposit of {amount} to account {account_number}.")

def withdraw_cash():
    st.subheader('Withdraw Cash')
    account_number = st.number_input('Enter Account Number', format="%.2f")
    amount = st.number_input('Enter Amount', step=0.01, format="%.2f")
    if st.button('Withdraw') and account_number and amount > 0:
        if account_number in accounts_df.index and accounts_df.loc[account_number, 'Balance'] >= amount:
            accounts_df.loc[account_number, 'Balance'] -= amount
            transaction_record = {'Date': datetime.now().isoformat(), 'Type': 'Withdrawal', 'From': account_number, 'To': None, 'Amount': amount}
            transactions_df.loc[len(transactions_df) + 1] = transaction_record
            save_data(accounts_df, ACCOUNTS_FILE)
            save_data(transactions_df, TRANSACTIONS_FILE)
            st.success('Withdrawal successful!')
            logging.info(f"Withdrawal of {amount} from account {account_number}.")
```

The above diagram show the implementation of some functionality of the banking system. It shows the operation of the cash withdrawal and deposition.

- Critical evaluation:

For the development of the banking system, there are many software tools which have played a vital role in development and implementation.

A] Advances in Software Processes

- Methodology: Agile methodology was adopted as it can work with the dynamic environment. Agile methodology specially the scrum, was chosen due to iterative nature and flexibility.
- Agile processes: the implementation of the agile process facilitated constant inspection, adaption and Improvement. Improved the product quality by addressing the issues.

B] Software engineering techniques

- Project management: JIRA is used to track and monitor the progress, tasks, and backlogs of the project.
- Prototype design: SKETCH tool is used to design and conceptualize the user interface. SKETCH is used to design the system design and workflow of the project.
- Version control: Code were managed and version controlled using the github.

C] How advanced Software Systems and Software Engineering have changed how we interact as a society with your system considering the following:

- Social impact: Online and mobile banking have made this financial services accessible virtually from any part of the world. The banking system has democratized the banking system accessible to all.
- Ethical impact: here, these project address the ethical concern by implementing the security and compliance feature by two factor authentication and data encryption.
- Entrepreneurial impact: the development of the banking app can get into entrepreneurial activity by providing start-ups and small business for managing finance effectively.

Works Cited

de Matos E. C. B., Moreira, A. M., & de Souza Neto, J. B. journal of the Brazilian Computer Society [Journal]. - 2016. - 22(8).

Delfim F. M., Paixão, K. V. R., Cassou, D., & Maia, M. de A journal of the Brazilian Computer Society [Book]. - 2018. - Vol. 22(9).