

Advanced Software Engineering

CMP9134M



STUDENT NAME:

STUDENT ID:

Table of Contents

[Section 1: Agile Methodologies and XP Techniques Employed 2](#_Toc161255745)

[1. Selection of Agile Methodology: 2](#_Toc161255746)

[2. User Stories and Planning: 2](#_Toc161255747)

[3. Simple Design and Pair Programming: 2](#_Toc161255748)

[4. Testing and Continuous Integration: 3](#_Toc161255749)

[5. Adaptation and Retrospectives: 3](#_Toc161255750)

[Section 2: Role, Tasks, and Contributions 3](#_Toc161255751)

[1. Role: 3](#_Toc161255752)

[2. Tasks and Contributions: 3](#_Toc161255753)

[Section 3: Project Planning 4](#_Toc161255754)

[1. Choice of Agile Framework - Scrum: 4](#_Toc161255755)

[2. Breaking Down the Work: 4](#_Toc161255756)

[3. Activities and Milestones: 5](#_Toc161255757)

[Section 4: Prototype Design and Implementation 5](#_Toc161255758)

[1. System Design: 5](#_Toc161255759)

[2. Interface Designs: 5](#_Toc161255760)

[3. Developed Product Evidence: 6](#_Toc161255761)

[4. Automation 6](#_Toc161255762)

[Section 5: Critical Evaluation of Software Engineering Tools and Techniques 6](#_Toc161255763)

[a. Advances in Software Processes: 6](#_Toc161255764)

[b. Software Engineering Techniques: 7](#_Toc161255765)

[c. Societal Impact of Advanced Software Systems: 7](#_Toc161255766)

Table of Figures

[Figure 1 ERD diagram of the bank management system 9](#_Toc161307912)

[Figure 2 Customer login page of the bank management system 9](#_Toc161307913)

[Figure 3 Home page of the bank management system 9](#_Toc161307914)

[Figure 4 Multiple user options 10](#_Toc161307915)

[Figure 5 Dummy user bank management options 10](#_Toc161307916)

[Figure 6 dummy user details 10](#_Toc161307917)

# Section 1: Agile Methodologies and XP Techniques Employed

While employing Agile methodologies, our banking project management system relied on various XP (Extreme Programming) practices to promote seamless interaction, ongoing optimisation, and precise delivery of a fine-tuned product by end of time. Below is a detailed description along with evidence of how our team utilized Agile methods and XP techniques:

## 1. Selection of Agile Methodology:

In the first stage of our project, the whole team determined that we would be employing the Scrum approach, a common Agile methodology which is known for its iterative approach and the main idea of close collaboration and adaptability (Alexandros et.al, 2017).

Evidence: Our initial team meeting proceedings concluded and the plan and roles were designated, including a Scrum Master responsible for facilitating scrum events and scrum principles.

## 2. User Stories and Planning:

We considered a wide range of inputs, including prioritization and implementation of user stories which includes features and functionalities of the end-users.

As for each user story, we calculated their complexity on multiple levels and team capacity were considered to choose what stories will be developed during sprints according to the project priorities.

Evidence: We use our Trello board as our project management tool which is divided into the project backbone of the user stories, prioritized by importance, as well as the sprint backlogs lists including the tasks for each sprint (Wautelet et.al, 2019).

## 3. Simple Design and Pair Programming:

During all development stages, we followed the XP approach with the emphasis on simple design.

Since pair programming sessions were implemented on regular basis, developers worked in group and made joint efforts to solve the same problems, sharing knowledge and offering immediate feedback on each other’s work to enhance the code quality.

Evidence: GitHub commit history and pull request reviews show the instanced of pair programming.

## 4. Testing and Continuous Integration:

With our development workflow linked to pytest framework, we integrated the automated testing due to which code changes did not introduce regressions or break existing functionality.  
Evidence: The directory in the bank automation system GitHub repository tests all the factors and functions using automatic test scripts.

## 5. Adaptation and Retrospectives:

Sprints reviews and retrospective of the team was held on a regular basis. The motive of these is to get the issues involved, what has been carried out successfully, what the team would correct or change in the future were discussed there.

Feedback taken from the retrospectives helped to transformed and optimized our development techniques, remove or cope with any impediments or challenges encountered.

Evidence: Meeting notes, and action points that were generated by our project documentation team show the team's ability to set aside time to reflect on what they did, review their performance, and learn from past actions (Marshburn, 2018).

# Section 2: Role, Tasks, and Contributions

Throughout the agile software development of the banking management system, I have got experience various positions in the team and have quite possibly been involved in various aspects of the project. Below is a detailed description of my role, tasks, and contributions:

## 1. Role:

Scrum Master: I was entitled with the role of a Scrum Master which involves facilitating Scrum meetings, fostering collaboration within the team, and ensuring the team follows the Agile principles.

## 2. Tasks and Contributions:

Facilitating Scrum Events: We used agile techniques and organized events such as sprint planning, daily stand-ups, sprint reviews, and retrospectives on a regular basis. In the conversations that I had with the rest of the team, I communicated clearly to my colleagues that they each had the chance to contribute, and we discussed our work progress and we even brought up the blocks.

Planning and Prioritization: I worked with PO and project team to prioritize the user stories, decompose them into small activities on the sprint backlog, and developed a time schedule. I lead the team in real-time discussions to ensure that everyone understood our sprint goals and would commit to acting and accomplishing our goals.

Ensuring Agile Principles: I promoted agile principles in a transparent manner and spearheaded processes around inspection and adaption. I made the team understand the virtue of change, the need for real collaboration among colleagues, as well as the importance of constant refinement of all our processes and practices.

Documentation and Reporting: I used project documentation very efficiently. Among the tools I used where meeting minutes, sprint backlogs and retrospective notes. Furthermore, I put together periodical progress reports and relayed intermediary project status to stakeholders with a view of maintaining transparency and consistency with the project objective.

Continuous Improvement: The retro meetings that I led for the team at the end of each sprint to look back and go over whether our performance matched plan was an important exercise and a great idea to the development process which helped us identify areas that could be improved upon. Team members learned how to be adaptable and continue to reflect, thereby improving their performance (Moraga & Piñango, 2023).

# Section 3: Project Planning

The plan of project planning stage was key for the implementation of the banking management system. We employed Agile methodology and practices to ensure our planning turned out to be flexible, iterative, and aligned with the project goals. Below, I will describe our approach to project planning, including our choice of the Scrum framework and how we broke down the work into activities and milestones:

## 1. Choice of Agile Framework - Scrum:

We chose the Scrum framework for its incremental approach that allow us to adapt to any changing requirements and deliver values to our stakeholders quickly.

## 2. Breaking Down the Work:

Backlog Refinement: First, we collected all the product requirements, user stories that represent the functionalities of the system into a single list called the product backlog.  
User Story Estimation: Each story got assigned a complexity level based on tools like story points. It not only enabled us to prioritize our tasks but also contributed to our appropriate planning of work.

Sprint Planning: Every time we had a planning sprint, we selected user stories from the product backlog and converted them into smaller, executable tasks.

Task Allocation: Function tasks were allotted to team members in accordance with their skills, expertise, and hours of availability. Every team member had clear knowledge concerning realizing the task and input at all levels of the sprint.

Daily Stand-ups: Every day standup meetings were conducted to trace progress, identify the blockages and modify the plan to outplay the barrier.

## 3. Activities and Milestones:

Sprint Goals: Every sprint goal or objective aimed at implementing a clickable version of a particular story around a defined Objective. Sprint objectives not only kept us be on track but also tackled the confusing thing by directing us towards what is necessary.

Sprint Reviews: At the end of each sprint, we had sprint reviews to demonstrate what has been completed and consult on how to optimize the product. By way of this approach, we got to have gaps found, criticism accepted and plans refactored for the better use of our efforts.

Retrospectives: At the end of each sprint, we had a retrospective meeting, which allowed us to analyse our process from the past sprint, figure out the stuff which were done properly and the things could be improved. We also came up with actionable improvements to those things that were not performed well in the past sprint (Al-Saqqa et.al, 2020).

# Section 4: Prototype Design and Implementation

In this part we will explain the prototype design for our banking management system that comprises the environment design.

## 1. System Design:

Architecture: The system follows a client-server architecture with the backend implemented in Python language and Flask framework on the server side as well as the SQLite database on the client side of our banking system infrastructure.  The frontend side has a ready-to-use interface from HTML, CSS, and JavaScript.

## 2. Interface Designs:

Main Page: Main banking system interface on the initial screen consists of controls for account management, financial operations, account service, and compliance/security.

Functionality Pages: We envision various use cases and design screen prototypes for operation, transaction record, inquiry, and security respectively. These designs prioritize user-friendly, simple, and with the best navigation ease.

## 3. Developed Product Evidence:

GitHub Repository: We have established a Git repository as a storage area for all the code used, as well as the documents, models, and test artifacts we worked on during the development process of our bank management system.

Demonstration Video: We have developed a short video highlighting our implemented banking functionalities of account creation, funds transfer, transaction history viewing and security settings configuration, to enable users employ the required feature.

## 4. Automation

Continuous Integration: Automated tests are integrated into our continuous integration pipeline using GitHub Actions, ensuring that tests are run automatically on every code push to the main branch. This facilitates early detection and resolution of issues, helping us maintain a robust and stable software product (Shahin et.al, 2017, pp.3909-3943).

Overall, our prototype design reflects our commitment to delivering a user-friendly and functional banking management system. The evidence provided demonstrates the development of the product and our implementation of automated testing to ensure its quality and reliability.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.no** | **Activity** | **Start Date** | **End Date** | **Total Days** |
| 1 | Project Planning | 22-02-2024 | 23-02-2024 | 1 |
| 2 | Git hub repository Setup | 23-02-2024 | 23-02-2024 | 1 |
| 3 | Sqlite3 database setup | 24-02-2024 | 24-02-2024 | 1 |
| 4 | Development of Account Management part of the banking system | 26-02-2024 | 28-02-2024 | 3 |
| 4.1 | Account Management  Design review sprint meeting | 26-02-2024 |  |  |
| 4.2 | Account Management  Design Testing | 27-02-2024 |  |  |
| 5 | Development of Financial Transactions part of the banking system | 29-02-2024 | 02-03-2024 | 3 |
| 5.1 | Financial Transaction  Design review sprint meeting | 01-03-2024 |  |  |
| 5.2 | Financial Transaction  Design Testing | 02-03-2024 |  |  |
| 6 | Development of the  Account Services of the banking system | 04-03-2024 | 07-03-2024 | 4 |
| 6.1 | Design review of the  developed service | 05-03-2024 |  |  |
| 6.2 | Developed design testing of the system | 06-03-2024 |  |  |
| 7 | Sprint Meetings to  overview the complete  design | 08-03-2024 | 10-03-2024 | 3 |
| 7.1 | Design modifications according to the feedback in sprint meeting | 09-03-2024 |  |  |
| 7.2 | Complete bank  management system  Testing | 10-03-2024 |  |  |

Project Chart

Source: (Self-created)

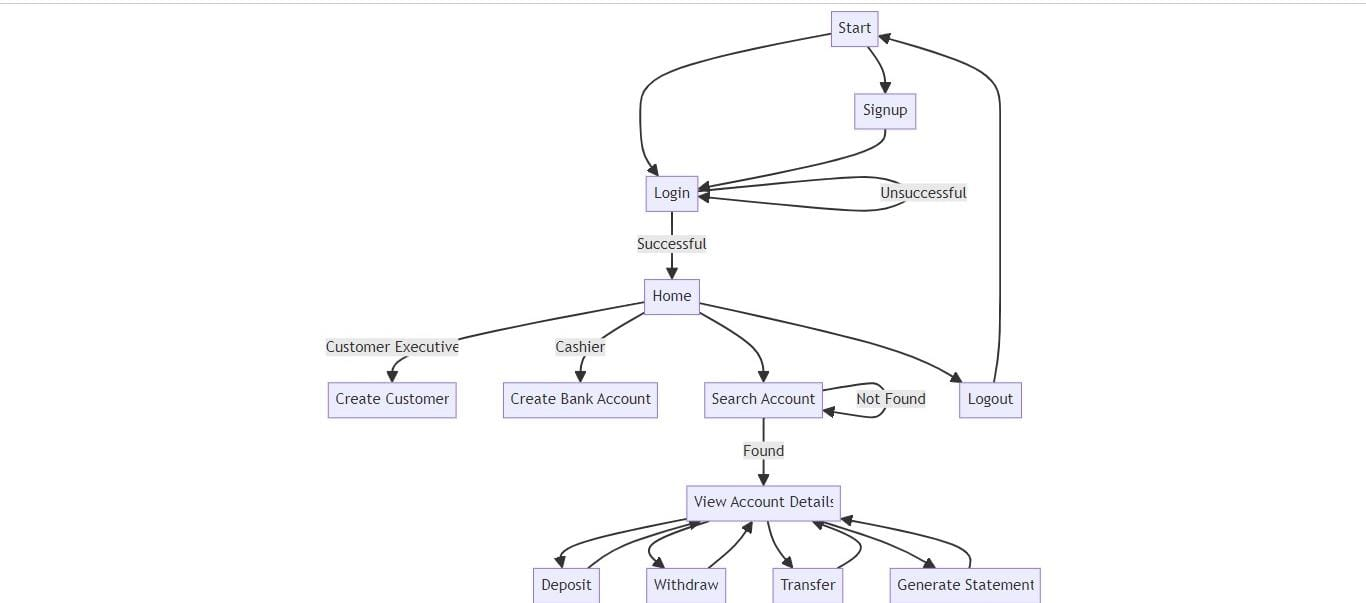


Figure 1 ERD diagram of the bank management system

Source: (Self-Created)

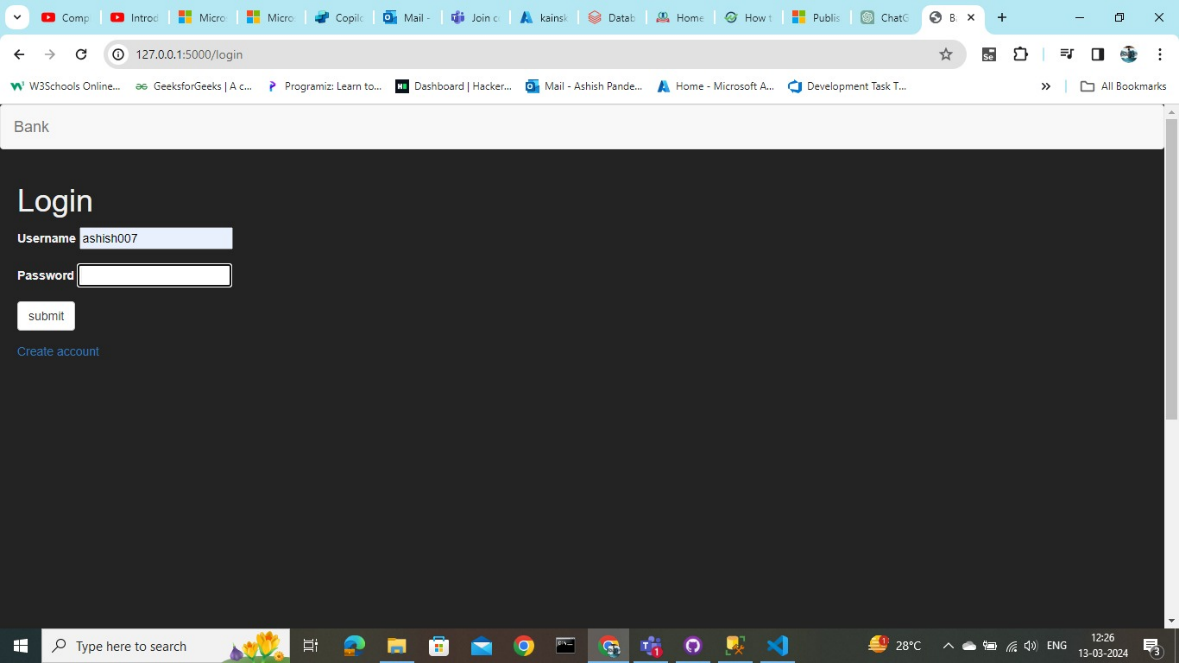


Figure 2 Customer login page of the bank management system

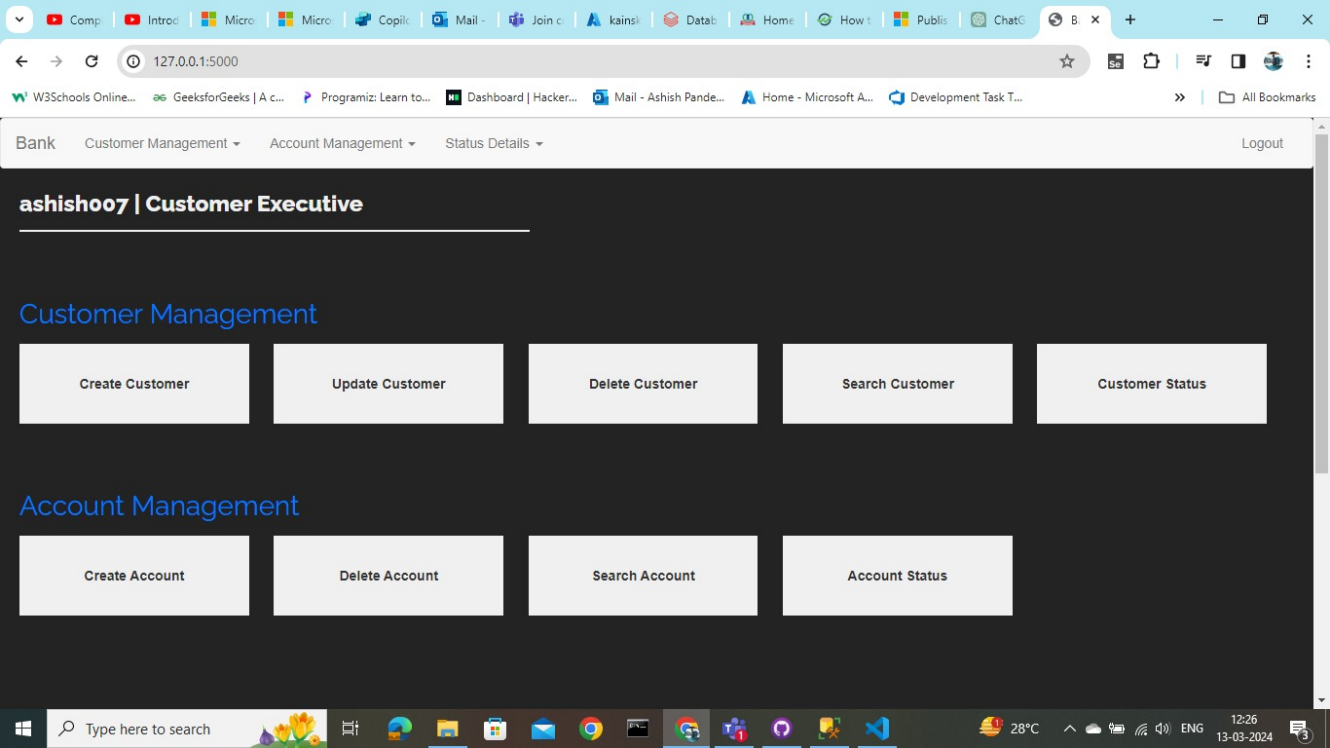


Figure 3 Home page of the bank management system

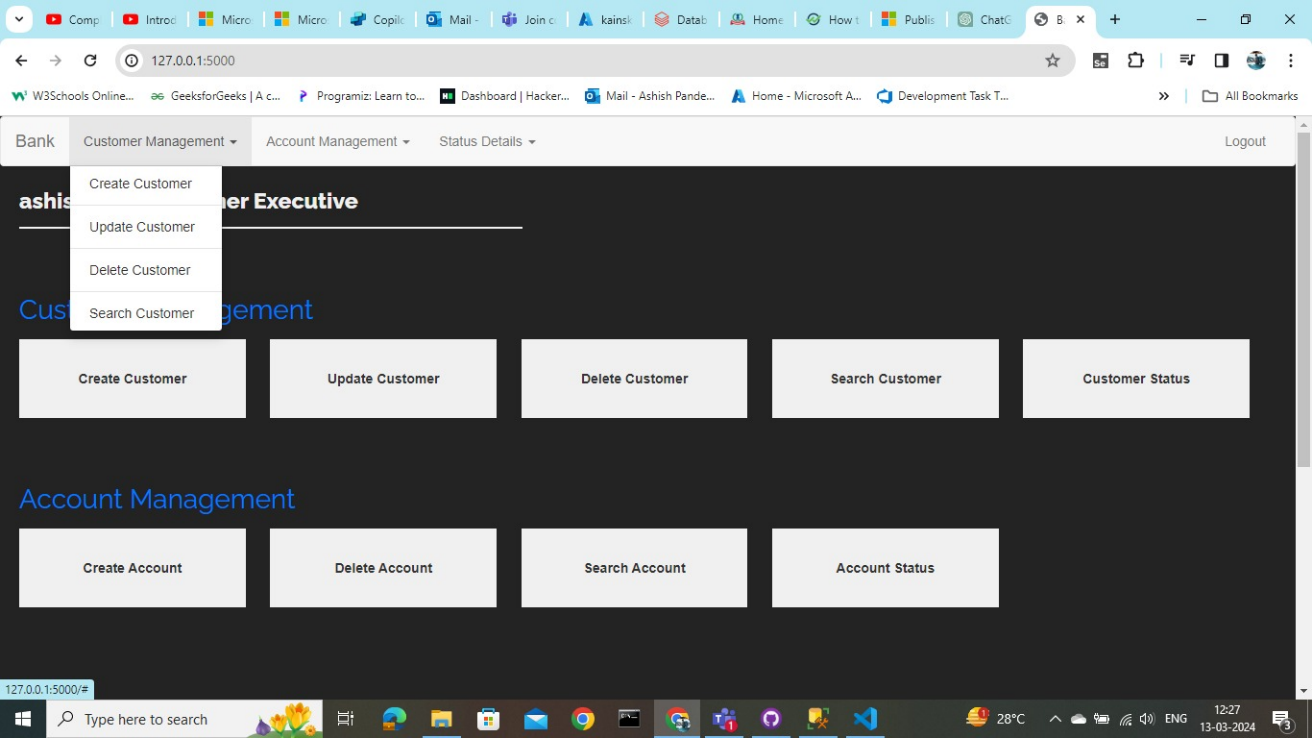


Figure 4 Multiple user options

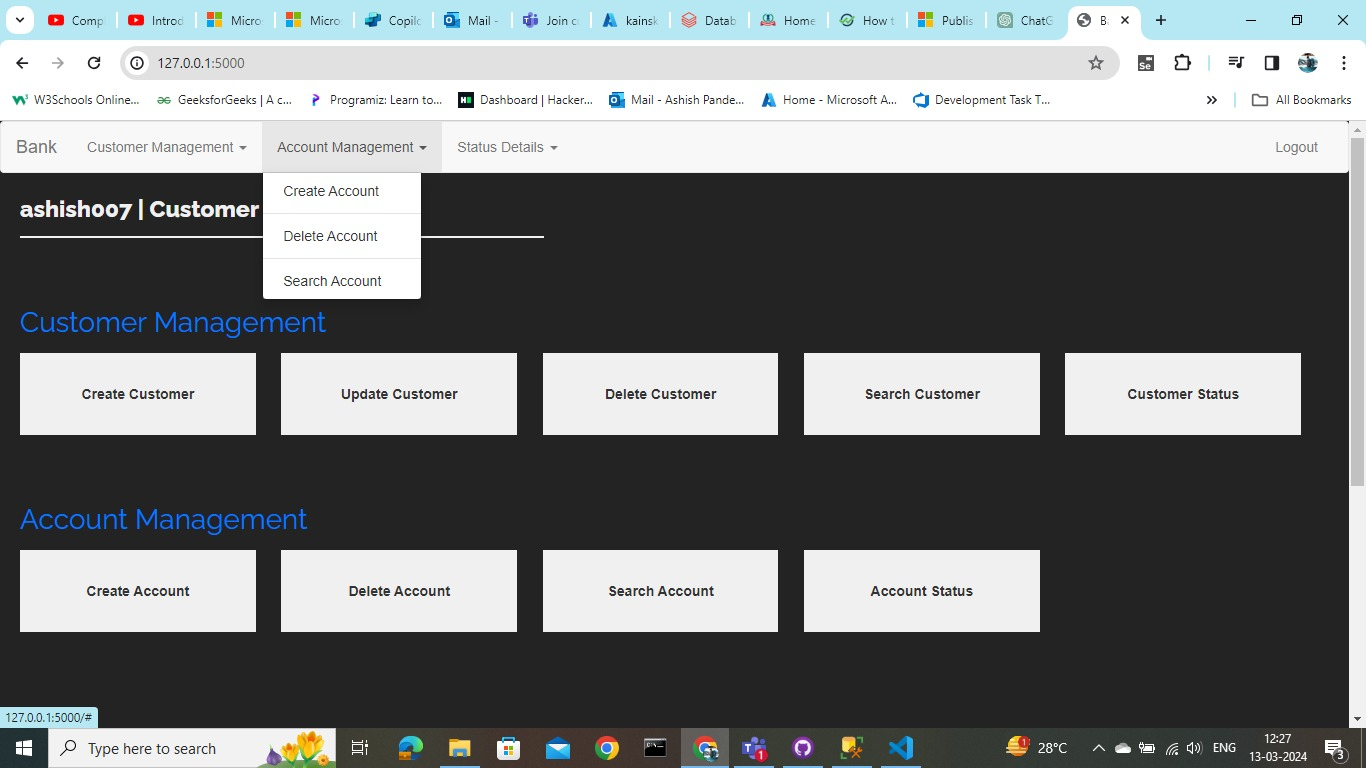


Figure 5 Dummy user bank management options

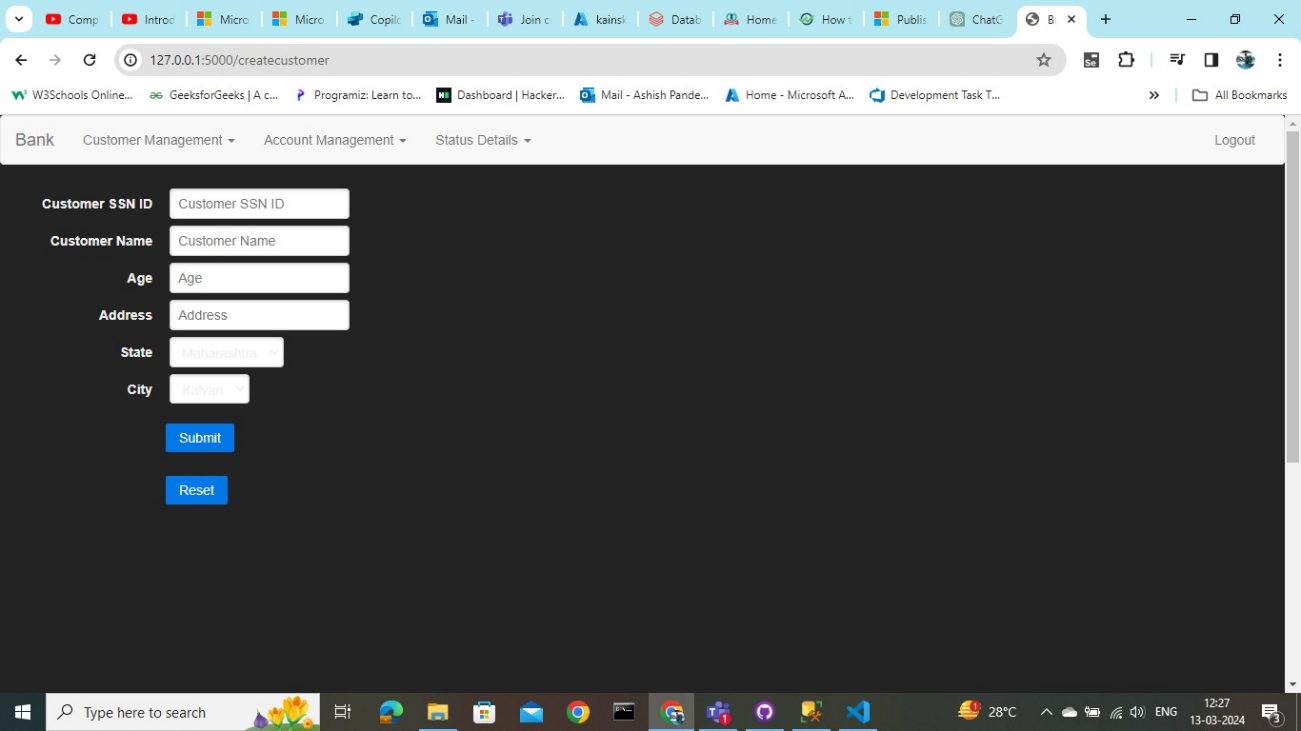


Figure 6 dummy user details

# Section 5: Critical Evaluation of Software Engineering Tools and Techniques

In this part we will evaluate the software engineering tools and methods used in the making of our bank management system, which include recent advances in software development processes, software engineering techniques, while also examining the positive and negative impacts of such advanced employing of software systems on the society.

## a. Advances in Software Processes:

**i. Methodologies:**

Evaluation: Scrum, was chosen as our Agile methodology, which brought us the perfect toolkit for iterative development and communication within the team. The methodology allows us to adjust to any changing expectations.

Evidence: It is through our adherence to the Scrum principles such as short sprint planning, frequent daily stand-ups, sprint reviews, and retrospective which are documented on the project management tool (Trello board).

**ii. Agile Processes:**

Evaluation: Agile methodologies helped us to respond quickly, with prompt implementation of the changes suggested, prioritizing of elements based on the value they provide to the users, and incremental delivery of the software (Pasha et.al, 2018, pp.12412-12424).

Evidence: User stories, deployment or retrospective of the sprint planning sessions and the continuous integration pipelines demonstrate our implementation of the principles and practices of an Agile working method.

## **b. Software Engineering Techniques**:

**i. Project Management:**

Evaluation: Agile methodologies played the important role in the success of our product and project management practices, such as backlog refinement, sprint planning, and regular retrospectives kept us focused, on track, and flexible during the whole development process (Andrei et.al, 2019).

Evidence: The Trello board tools contains the open backlogs, sprint plans, and notes from retrospectives, which is responsible to showcase our systematic approach to managing projects.

**ii. Prototype Design:**

Evaluation: Prototyping in our case was crucial for drafting the system architecture and the interface design that later allowed us to communicate efficiently and interact with the stakeholders and gave a direction to the whole development process (Schuh et.al, 2018).

Evidence: The mock-ups of screens and system diagrams were presented in such a way that indicating the type of user interface and how the system functions. This has been aided in the generation and validation of requirements.

**iii. Version Control:**

Evaluation: GitHub was used as a version control platform that allowed collaborative development, code management, and tracking of changes over the entire period of project execution stage.

## **c. Societal Impact of Advanced Software Systems**:

**i. Social Impact:**

Evaluation: Banking management system reduces the hassles of physically visiting bank or so and makes the transactions easy and prompt for users, consequently promoting financial inclusion & might give a sense of empowerment.

Evidence: The user feedback and acceptance estimation will enable to analyse the system's impact on how people behave and also impact society as a whole.

**ii. Ethical Impact:**

Evaluation: Another aspect of ethical implications is data safety, privacy and fairness because they happen at the moment of the designing and implementation of such structure to make the system trustworthy, responsible, and regulated.

Evidence: Adoption of preventive measures, practices of encrypting data, and data protection compliance manifests that we are not neglecting the ethos and privacy of users.

**iii. Entrepreneurial Impact:**

Evaluation: The banks system opens up the opportunities for the development of the fintech startups and businesses to introduce the innovative practices, have a positive differentiation and create the value-added services in the financial sphere.

Evidence: Through market analysis, getting feedback on customers, and adoption within the system, market potential for a system and entrepreneurial viability may be assessed so to help business development and innovation (Toms et.al, 2020, pp.105-121).

In conclusion, the implementation of the software engineering tools and approaches in our software development for the banking management system has contributed to a finely-tuned, user-centric software product that takes into consideration societal impacts and ethical considerations. Consistent upgrading and adaptation of software systems and technologies will be the most important factors in the fulfilment of user needs and the irresistible trend of the society.

# References

Alexandros, N.K., Sakas, D.P., Vlachos, D.S. and Dimitrios, N.K., 2017. Comparing scrum and XP agile methodologies using dynamic simulation modeling. In *Strategic Innovative Marketing: 5th IC-SIM, Athens, Greece 2016* (pp. 391-397). Springer International Publishing.

Al-Saqqa, S., Sawalha, S. and AbdelNabi, H., 2020. Agile software development: Methodologies and trends. *International Journal of Interactive Mobile Technologies*, *14*(11).

Andrei, B.A., Casu-Pop, A.C., Gheorghe, S.C. and Boiangiu, C.A., 2019. A study on using waterfall and agile methods in software project management. *Journal of Information Systems & Operations Management*, pp.125-135.

Marshburn, D., 2018. Scrum retrospectives: Measuring and improving effectiveness.

Moraga, R. and Piñango, H., 2023. Continuous Improvement in Software Development and Digital Product Management: Addressing the Challenges of the Digital Economy.

Pasha, M., Qaiser, G. and Pasha, U., 2018. A critical analysis of software risk management techniques in large scale systems. *IEEE Access*, *6*, pp.12412-12424.

Schuh, G., Doelle, C. and Schloesser, S., 2018. Agile Prototyping for technical systems–Towards an adaption of the Minimum Viable Product principle. *DS 91: Proceedings of NordDesign 2018, Linköping, Sweden, 14th-17th August 2018*.

Shahin, M., Babar, M.A. and Zhu, L., 2017. Continuous integration, delivery and deployment: a systematic review on approaches, tools, challenges and practices. *IEEE access*, *5*, pp.3909-3943.

Toms, S., Wilson, N. and Wright, M., 2020. Innovation, intermediation, and the nature of entrepreneurship: A historical perspective. *Strategic Entrepreneurship Journal*, *14*(1), pp.105-121.

Wautelet, Y., Gielis, D., Poelmans, S. and Heng, S., 2019. Evaluating the impact of user stories quality on the ability to understand and structure requirements. In *The Practice of Enterprise Modeling: 12th IFIP Working Conference, PoEM 2019, Luxembourg, Luxembourg, November 27–29, 2019, Proceedings 12* (pp. 3-19). Springer International Publishing.