

ATM 2.0

Sprint Three - Planning Document

Team 25

Anthony Goeckner

Krutarth Rao

Harold Smith

Austin Reed

Scrum Master: Harold Smith

Meeting Schedule: Tuesday, Thursday & Sunday - 3:00 p.m. to 4:30 p.m.

Contents

[Contents](#)

[Overview](#)

[Challenges](#)

[Research](#)

[User Story #1](#)

[Tasks](#)

[Acceptance criteria](#)

[User Story #2](#)

[Tasks](#)

[Acceptance criteria](#)

[User Story #3](#)

[Tasks](#)

[Acceptance criteria](#)

[User Story #4](#)

[Tasks](#)

[Acceptance criteria](#)

[User Story #5](#)

[Tasks](#)

[Acceptance criteria](#)

[User Story #6](#)

[Tasks](#)

[Acceptance criteria](#)

[User Story #7](#)

[Tasks](#)

[Acceptance criteria](#)

[Remaining Backlog](#)

[As a User](#)

[As a Customer](#)

[As a Developer](#)

Overview

The theme for sprint three will be to merge the core functionality features that each member has been working on individually. So far, the GUI, the network communication, interaction with the biometric hardware devices and facial recognition are complete along with the targets mentioned in the planning document for the first sprint.

The team will focus on calling the API that other members have written in order to construct a fully functional product that entails effective communication among core units to provide the user with a secure and user-friendly ATM system. There is more to be done in this sprint than in the previous sprint, because we underestimated the overall workload. We hope to overcome this by completing tasks ahead of schedule by devoting more hours than in previous sprints.

Challenges

The major challenge of this sprint will be connecting the large sections of the ATM and ensuring that every feature is implemented. This will involve learning each others contribution to the ATM, how to work with their code. This sprint requires more work than the previous sprints, because of this we will need to be more productive. We have been working the most towards the later half of our sprints, we expect to complete our tasks sooner into the sprint so that we can ensure that the unexpected problems are solved.

Research

Apart from meetings, each member is responsible for reading and understanding relevant documentation to contribute effectively in the project. To that end, each member will spend a reasonable amount of time researching the topics that are related to their portion of the project.

Krutarth

- Relational databases and querying (4 hours)
- Optimizing local image storage (2 hours)
- Card swipe device (3 hours)

Anthony

- Fingerprint matching libraries (4 hours)
- VPN systems (3 hours)

Harold

- Fingerprint Authentication Libraries(4 hours)

- SSL implementation (1 hour)

Austin

- VPN (5 hours)
- Database (5 hours)

User Story #1

“As a developer, I would like the database communicator class to generate queries in real time to look-up, update or delete entries in the database.”

Implementing this user story ensures that the authentication class object can authenticate users using a database communicator object. *User story one is carried forward from sprint 2 because a goal with higher priority was achieved by the team during the allotted time.*

Tasks

- Write test cases to check output from querying function. (2 hours, Rao)
- Complete Query and Delete methods in database communicator class. (5 hours, Rao)
- Identify and match the callbacks necessary for each return value from database. (4 hours, Rao)

Acceptance criteria

- Given an active server program when a database is loaded then the program should be able to connect to the database.
- Given an active server program when the program queries the database then database able to serve a query request and return valid data.
- Given an active server program when queries are requested at high frequency then the database should be able to serve each query without substantial latency.
- Given a completed authentication when an authentication object attempts to authenticate a user, the object should be able to receive output from the database in a reasonable time frame.
-

User Story #2

“As a developer, I would like the graphical user interface to be in communication with the rest of the client”

The goal of this is to finish the work of the last two sprints making one system. It will require implementing the logic of the GUI by merging the GUI project with the main Client.

Tasks

- Merge GUI project with Client (3 hours, Reed)
- Integrate Client-Server communication with GUI (1 hours, Reed)
- Design test cases for the integrated system (1 hours. Reed)

Acceptance criteria

- Given that the GUI and client are functional, then they should interact with each other.
- Given that the GUI is within the client structure, then the client should control the state logic of the GUI.
- Given that the GUI and client are communicating, then the GUI should be able to send data to and display data from the server using the client controller and server connection classes as mediators.

User Story #3

“As a developer, I need to have full functionality of the GUI”

“As a user:

I need to withdraw specific amounts of money.

I need to deposit money.

I need to view my account balance.

I need to manage my security preferences, such as enabling/disabling biometric authentication.”

Tasks

- Create the visual aspect of the GUI (4 hours, Reed)
- Develop the functionality of the GUI(5 hours, Reed)
- Implement Tests for new features (3 hours. Reed)

Acceptance criteria

- Given that the user is logged into an account, then every expected functionality should be displayed for them to select.
- Given that a user wishes to withdraw money, deposit money, view account, or manage settings, then the complete functionality will process that request.
- Given that a user interacts with the ATM, then they have a user friendly experience.

User Story #4

“As a developer, I would like to store user’s fingerprint and facial recognition images on the server in a secure manner.”

Due to the nature of the image matching algorithms for fingerprints and facial recognition, we need to store the verification images locally on the machine running the algorithm.

Tasks

- Collect the verification images beforehand during account creation. (2 hours, Rao & Smith)
- Encrypt the directory the stores the base images and develop efficient decryption technique (6 hours, Rao & Smith)
- Store local image file paths in database. (1 hours, Rao)
- Test images by passing through the verification algorithm (2 hours. Rao)

Acceptance criteria

- Given that there are multiple images of the same person, the images are stored in a uniform and retrievable fashion.
- Given that an image is captured, then the images should be stored in an encrypted folder only accessible by the server application.
- Given that a base image is collected, then the data should be stored and formatted for use by other parts of the system.

User Story #5

“As a developer, I would like to integrate the authentication module with the networking module for client-server-database communication cycle to take place”

The communication between system modules to authenticate the user and accommodate transaction requires that the authentication classes are calling the correct API from the networking module.

Tasks

- Establish the networking functionality needed by the authentication classes (2 hours, Rao)
- Identify the networking API needed in addition to present implementation (2 hours, Rao)
- Call the needed networking calls to initiate the TCP-Authentication class communication(5 hours. Rao & Smith)

Acceptance criteria

- Given that the user of the ATM initiates an action that requires authentication, then the client will communicate with the server through the TCP and authentication classes to permit further use.
- Given that the authentication logic is traversed, then the correct result for the authentication should be passed along to the client through the networking module.

User Story #6

“As a developer, I would like the connection between the client and the server to occur with a SSL encryption”

The networking module is what exposes the sensitive data to devices that are not in the developer's control. The developer has no method to determine the vulnerabilities that exist in the switches and routers that are intermediate nodes in the client-server communication. We overcome this security hazard by using SSL encryption. SSL is an additional layer to the VPN that provides security in addition to the end-to-end encryption technique.

Tasks

- Introduce SSL encryption in all outgoing communication. (3 hours, Rao & Goeckner)
- Receive SSL encrypted traffic. (3 hours, Rao & Smith)
- Test active SSL through packet flags. (1 hour, Goeckner)
- Attempt to snoop packets in the network and confirm encryption(5 hours, Rao)

Acceptance criteria

- Given that a TCP connection between the server and client can be established, then all packets sent and received should be SSL encrypted.
- Given that the traffic is encrypted when leaving the client, then the server should be able to decrypt the .
- VPN uses key-based security.
- Connection through VPN is successful and traffic reaches intended destination.

User Story #7

“As a developer, I need to compare two fingerprints to verify the user’s identity.”

This user story details the implementation of a fingerprint verification library to be used on the server for authenticating users.

Tasks

- Find a library to perform fingerprint matching. (3 hours, Smith)
- Integrate library with ATM server application. (5 hours, Smith & Goeckner)
- Implement fingerprint verification using one fingerprint sent from ATM client and one stored on the server. (6 hours, Goeckner)
- Test fingerprint verification system. (4 hours, Goeckner & Smith)

Acceptance criteria

- Given that both fingerprint images are of the same finger, the system should register a success.
- Given that both fingerprint images are not of the same finger, the system should register a failure.
- Given that a fingerprint cannot be matched with confidence, the system should register a failure.

User Story #8

“As a user, I should start my session by swiping my card or entering the card number before authentication occurs.”

This user story details the implementation of user sessions and the authentication of card numbers and PINs.

Tasks

- Try working with “Square” card reader. (3 hours, Goeckner)
- If square does not work, implement manual card number entry. (3 hours, Reed & Goeckner)
- Set up authentication of card number and PIN. (4 hours, Goeckner)

Acceptance criteria

- Given a card number, the system should check if the account is in the database.

- Given that the account is in the database, the system should check that the PIN matches that found in the database.
- Given that developers choose not to use the Square reader, numbers should be input manually via the touch screen.

Remaining Backlog

The following requirements are yet to be implemented and were proposed as plausible extras.

As a User

- I need to manage my security preferences, such as enabling/disabling biometric authentication.
- I should be able to enter a backup password received from the bank in case I forget my PIN. (If time allows.)
- I would like the ability to enter an “emergency/duress PIN”, which will silently contact the police in case of a robbery. (If time allows.)

As a Customer

- I would like the machine to be capable of detecting sensor failures.
- I would like for my users to be authenticated a second time before allowing them to change their settings.
- I would like optional remote terminal access to the machine.

As a Developer

- I would like USB ports disabled to prevent unauthorized access.
- I would like to create an interface for the bank to add/modify customers’ details in the customer database. (If time allows.)
- I would like to have code that is sufficiently documented and well formatted. The software should be documented using a tool such as Doxygen, which generates documentation from comments placed in the code.
- I would like code that is modular and built to accommodate updates in the near or far future. Since the ATM 2.0 may be used by multiple, independent financial institutions, the system and tools must be reusable and modular to allow for easy application to new customers.