



CreditWatch

SOFTWARE ENGINEERING

REPORT

Written by Team 3:
Muhammad Ali, Hannah Beene, Justin Eggers, David Murray, Thao My Tran, Heiton Zang

Table of Contents

Software Engineering Report	1
Requirements and System Modeling	2
Approach to requirements	2
Functional Requirements	2
Non-Functional Requirements	3
Site Map	4
Use Cases	5
Wire Frames	11
Architecture Design	17
Implementation.....	19
Software Testing	19
Test Cases:.....	20
Executed Test Cases:	31
Requirements Traceability Matrix	33
Deployment.....	34
Glossary of Terms	i

Software Engineering Report

This document presents the solution engineered by our development team through the software life cycle, containing relevant information for banks and credit card companies looking to track their customers' unauthorized purchases. Our solution includes three main components: a user interface, database, and fraud detection process.

Requirements and System Modeling

Approach to requirements

The CreditWatchⁱ software will find abnormal patterns within people's purchases to detect potential credit card fraudⁱⁱ. CreditWatch will be an application credit card issuers can use to analyze their databases to make the analysis easier. The focus of our functional requirements was made considering we will have to sell a product to customersⁱⁱⁱ. We created our requirements to make sure the customers have an accessible place to manage and view the data they put into CreditWatch. Our non-functional requirements were made considering that our application needs to run efficiently while also making sure it is accurate and secure.

Functional Requirements

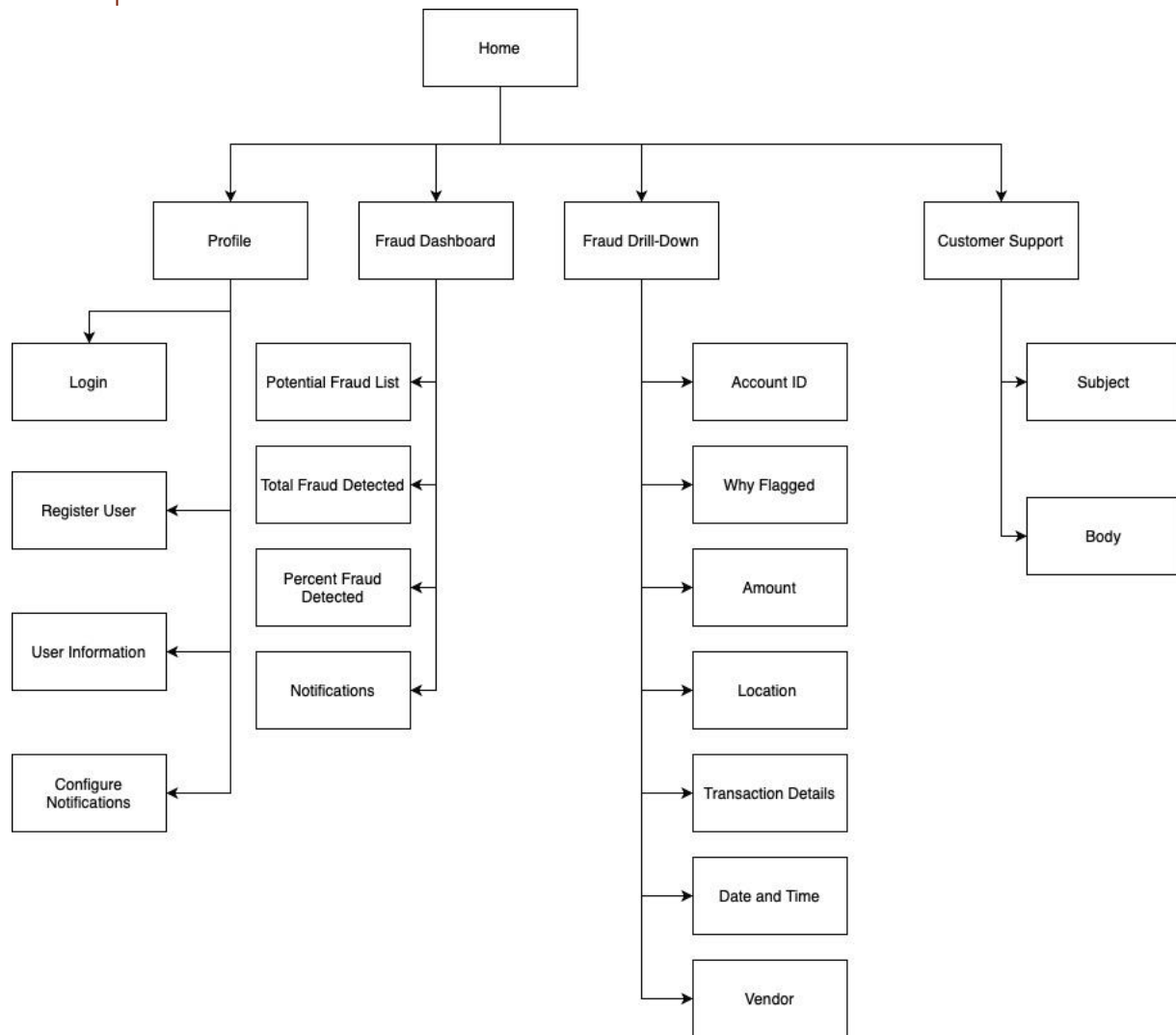
- | | |
|--|----------|
| R.1. Customers shall be able to manage user accounts ^{iv} | Hannah |
| 1. Customer shall be able to create a user account | Heiton |
| 2. Customer shall be able to edit user account details | David |
| 3. Specify user access ^v | Thao My |
| 4. Change security questions | Thao My |
| 5. Save billing info | Heiton |
| 6. Change billing info | Muhammad |
| 7. Specify user works for | Thao My |
| 8. Save user contact info | Thao My |
| 9. Verify user contact info | Heiton |
| 10. Edit username | Thao My |
| 11. Edit password | Thao My |
| 12. Save password | Thao My |
| i. Verify password meets rules | David |
| 13. Delete user accounts | Thao My |
| R.2. CreditWatch shall generate temporary passwords | David |
| 1. Automatic sending temp password with account creation | Justin |
| 2. Deactivate password after x time | David |
| 3. When signing in with temporary password ask for new password | Justin |
| R.3. Users shall be allowed to login | Hannah |
| 1. Verify password and username combo | David |
| 2. Reset password | Heiton |
| 3. Use security questions | Justin |
| R.4. Users shall be able to contact customer support | Hannah |
| 1. Track customer issues | Justin |
| 2. Respond to customer issues | Justin |

R.5. CreditWatch shall process billing	Hannah
1. Send receipts to customers	Justin
2. Track payments made and due	Justin
3. Charge for services ^{vi} according to time and amount of data	Hannah
4. Stop services when bills are overdue	David
R.6. CreditWatch shall detect fraud	Hannah
1. Import data	Hannah
2. Export results	Hannah
3. Search data for potential fraud	Hannah
4. Learn from past data	Hannah
R.7. CreditWatch shall protect data	Hannah
R.8. CreditWatch shall send fraud detected notifications	Heiton
R.9. CreditWatch shall send notifications about maintenance ^{vii}	David
R.10. CreditWatch shall track logistics ^{viii}	Hannah
1. Track total company users	Heiton
2. Track total accounts being processed	Heiton
3. Calculate total fraud detected ^{ix}	Heiton
4. Calculate percentage of users total at risk	Hannah
5. Calculate percentage of users at each specific company at risk	Hannah

Non-Functional Requirements

R.1. Process quickly	Hannah
R.2. Process in real time	Hannah
R.3. High up time	Hannah
R.4. Low down time	Hannah
R.5. Retain data for 3 years	Hannah
R.6. Few false or missed fraud notifications	Hannah
R.7. Low website load time	Hannah
R.8. Working load time	Hannah
R.9. High percentage of success	Hannah
R.10. Welcome screen	Hannah
R.11. Working links	Hannah

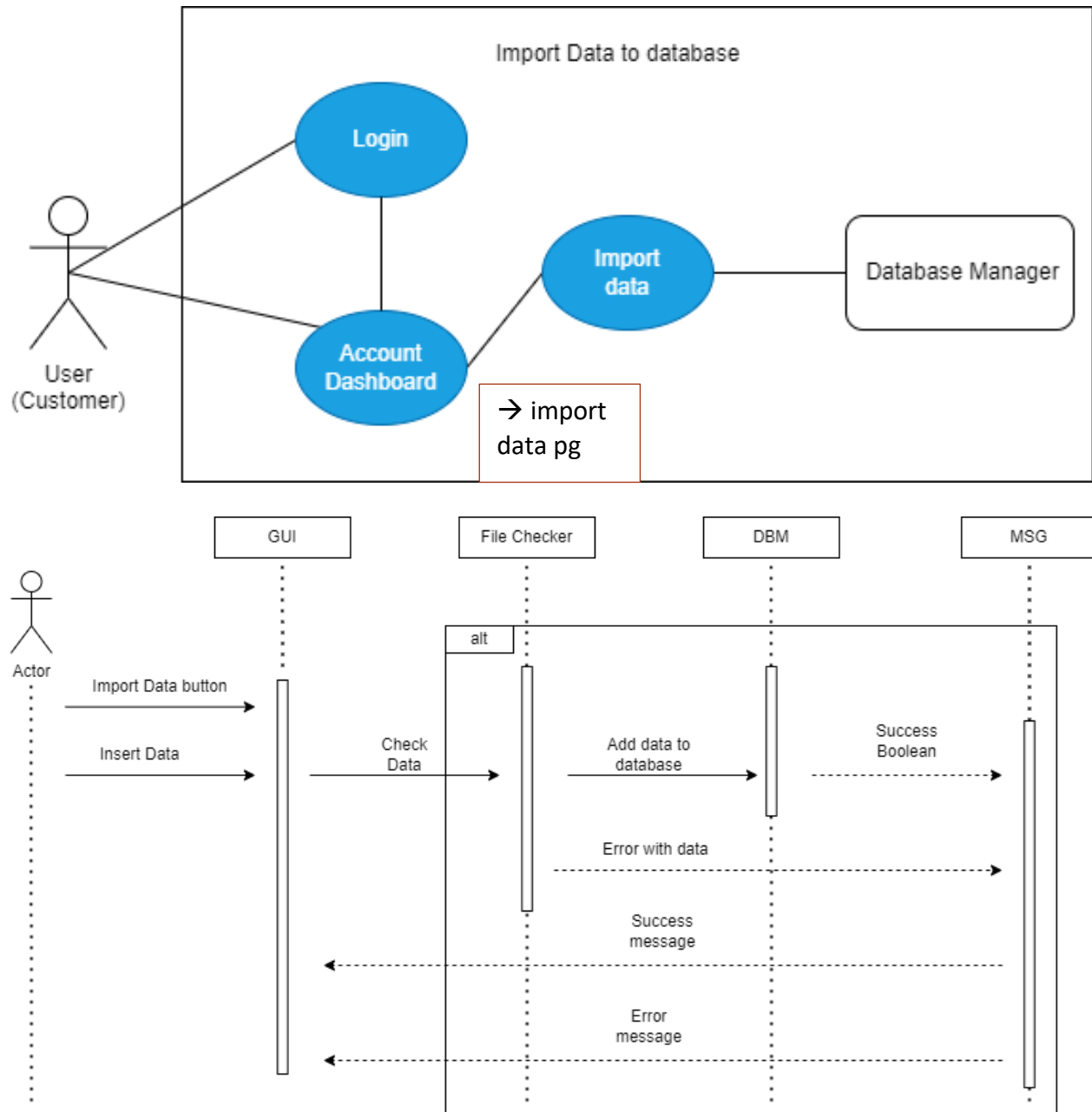
Site Map



Use Cases

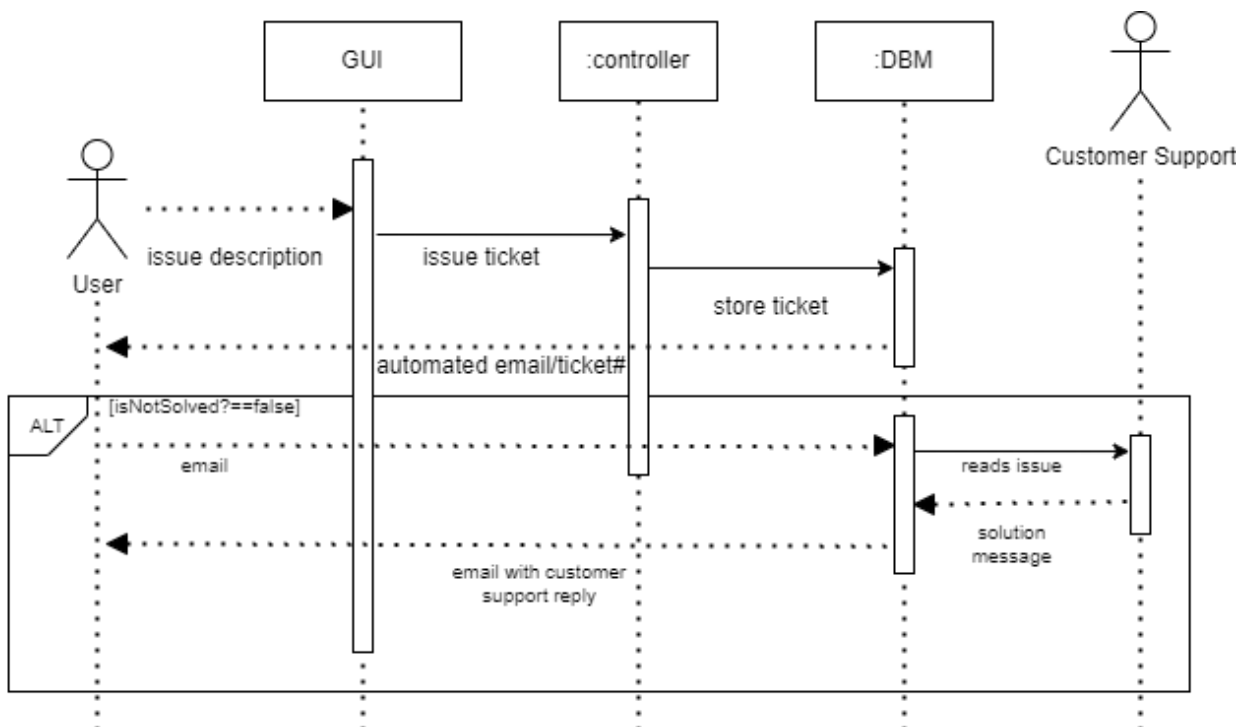
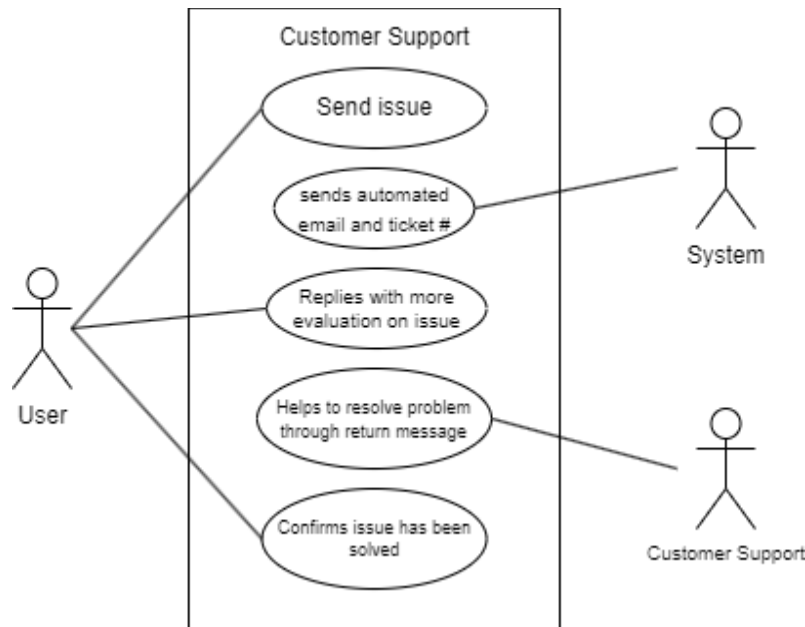
Import Data

Justin

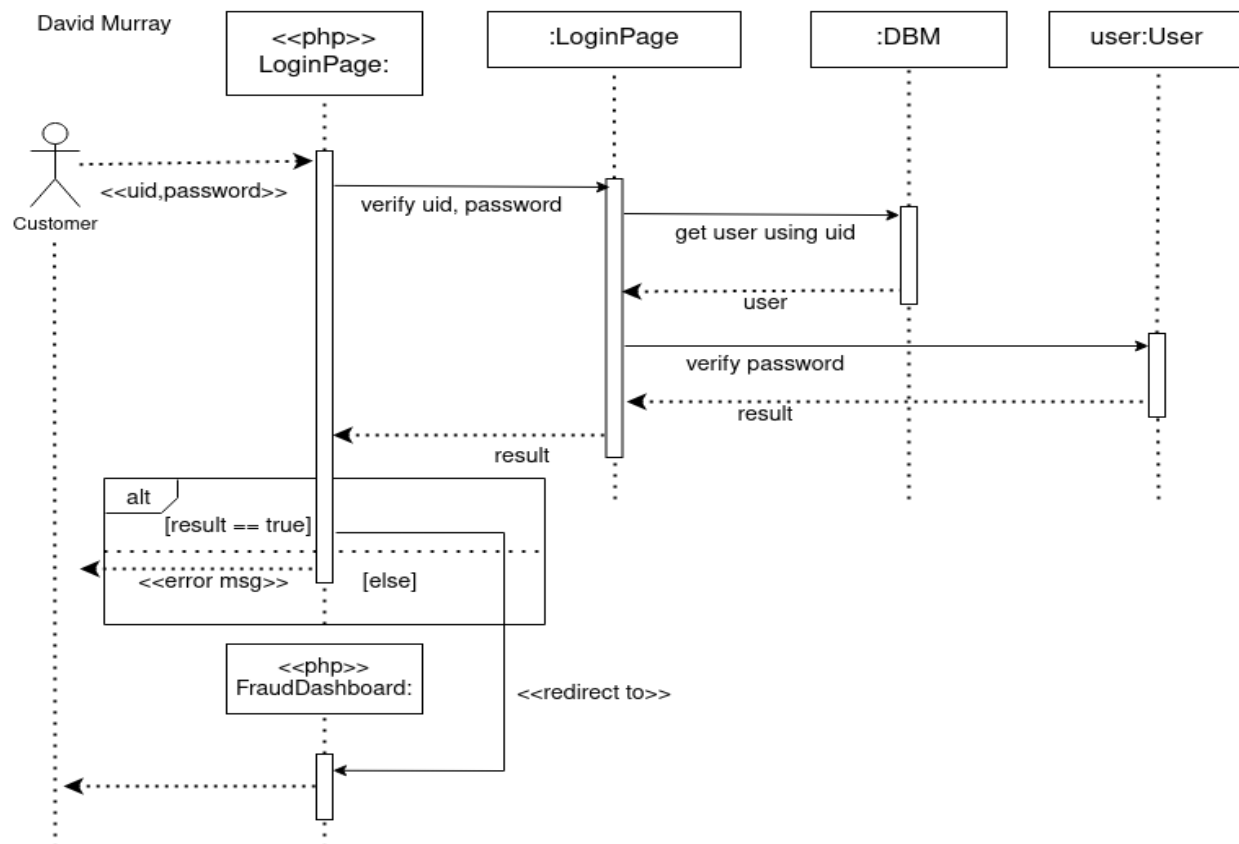
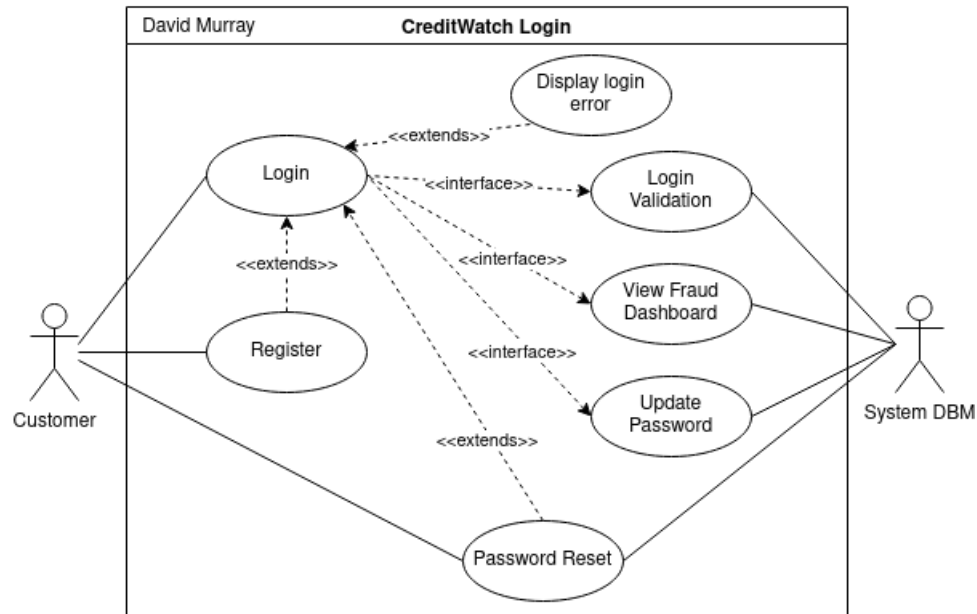


Contact Customer Support

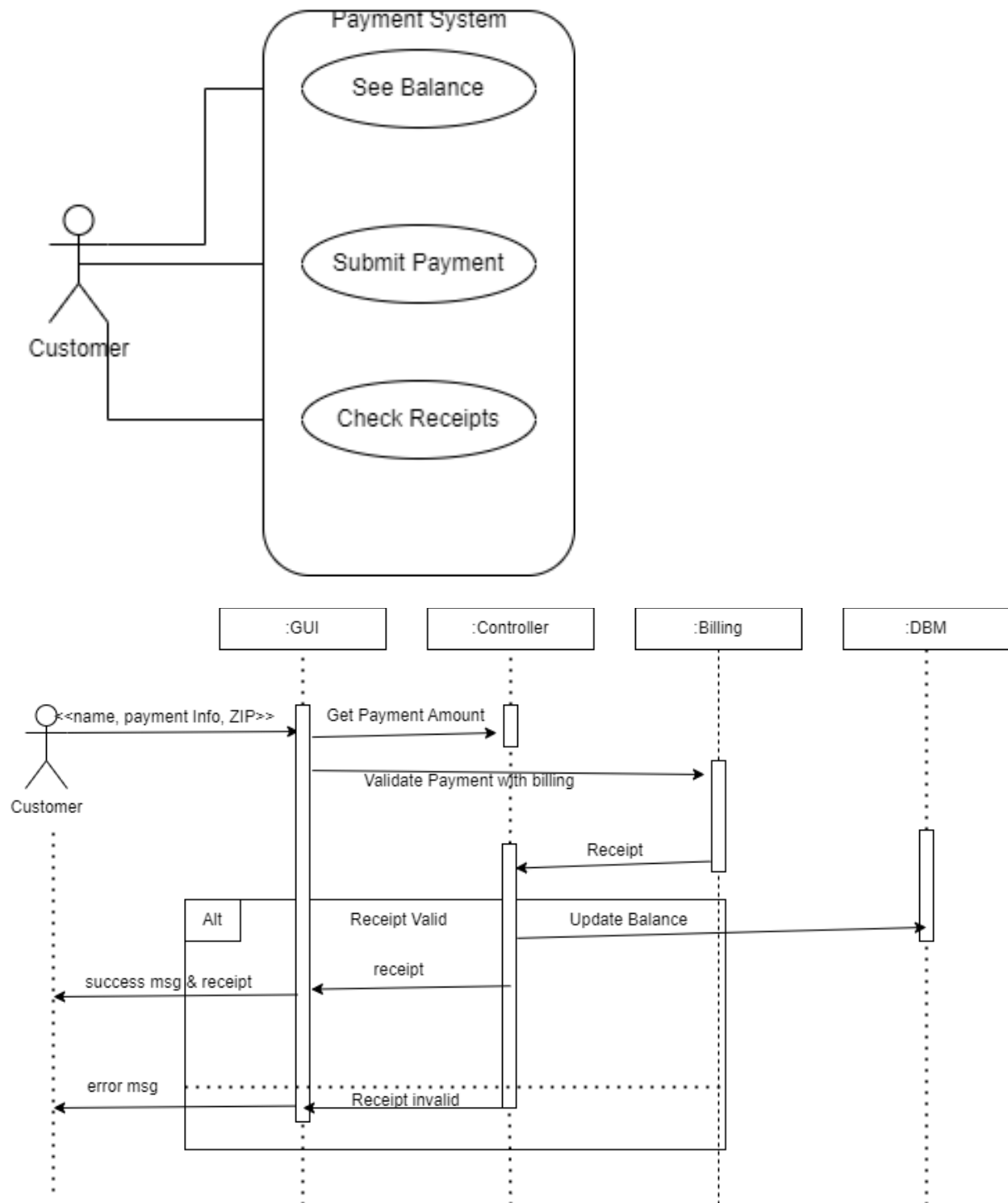
Muhammad



Login
David

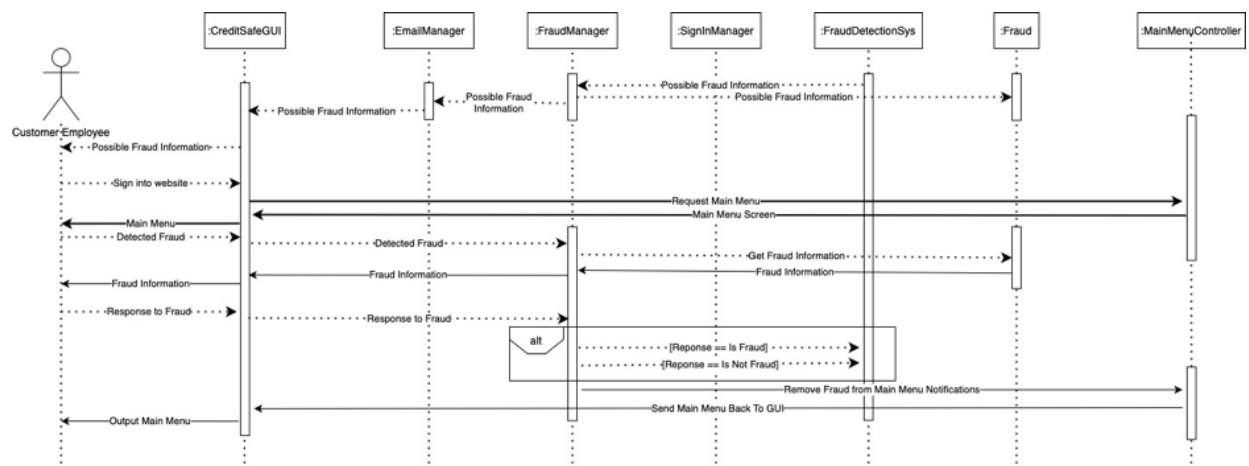
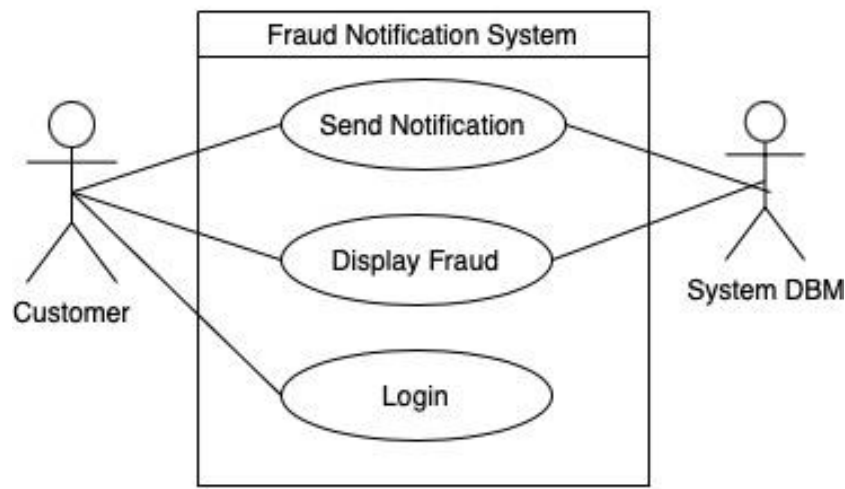


Paying Balance
Heiton



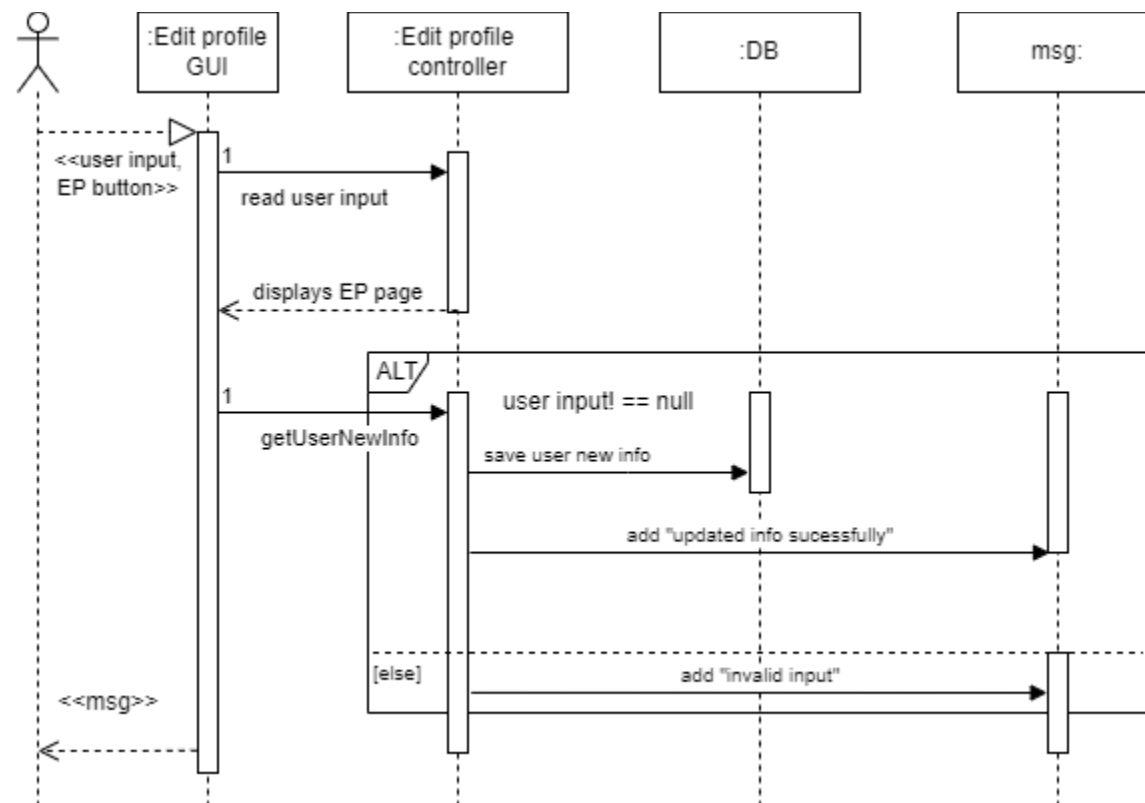
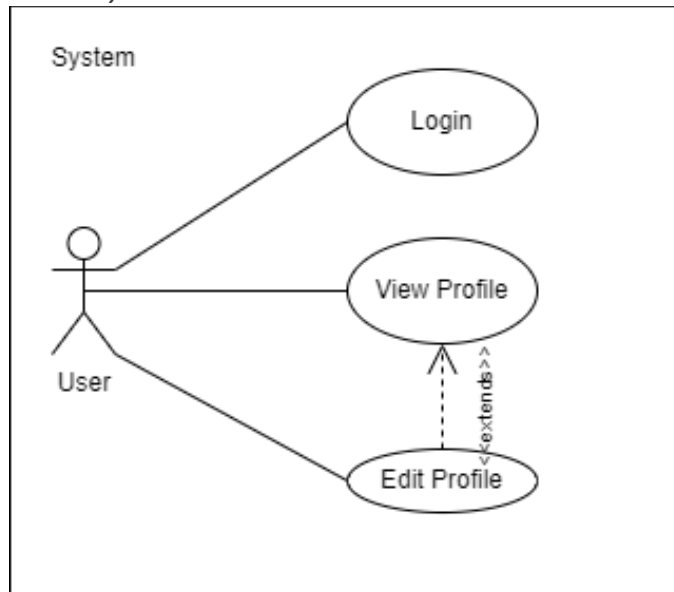
Notify User of Potential Fraud

Hannah



Edit Profile

Thao My



Wire Frames

Upload Data

David

CreditWatch

Home

Fraud Dashboard

Notifications

Upload Data

Payment

Profile

UserName

Dashboard

Profile

Change Password

Logout

Browse

Upload File

File selected: FILENAME.csv

File Sent: FILENAME.csv

File Size: # bytes

File type: text/csv

Login

David

CreditWatch

Home

Login

Register

Contact Us

Username

Password

Login

Don't have an account? Sign up now.

Contact Us

Muhammad

CreditWatch

[Home](#)[Profile](#)[Fraud](#)[Contact Us](#)[Logout](#)

Contact Us

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Iste quaerat autem corrupti asperiores accusantium et fuga! Facere excepturi, quo eos, nobis doloremque dolor labore expedita illum iusto, aut repellat fuga!

34 Street Name, City Name Here,
United States
+1 (222) 345 6789
info@mywebsite.com

Name

Subject

Message

Attachments



Make a Payment

Muhammad

Review Transaction

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Iste quaerat autem corrupti asperiores accusantium et fuga!

Lorem ipsum dolor sit amet, consectetur adipisicing elit.

Amount Due

Total before tax: \$ ####

Tax: \$ ####

Subtotal: \$ ####

☐ Payment Type 1: Card

Name:

Card Number:

Expiration Date:

 /

CVV:

CANCEL☐ Payment Type 2: Electronic Check

Front:



Back:

**SUBMIT
PAYMENT**

Payment History

Justin

CreditWatch

[Home](#)
[Fraud Dashboard](#)
[Notifications](#)
[Upload Data](#)
[Profile](#)
[Contact Us](#)

Payment History
Profile > Payment History

Product Name	Payment Date	Amount

Time till next Payment
Date:

UserName
Dashboard
Profile
Change Password
Logout

Reset Security Questions

Justin

CreditWatch

[Home](#)
[Fraud Dashboard](#)
[Notifications](#)
[Upload Data](#)
[Profile](#)
[Contact Us](#)

Reset Security Questions
Profile > Security > Security Questions

UserName
Dashboard
Profile
Change Password
Logout

Notifications
Heiton

Icon

Notifications

Dashboard after login

Notifications

Clear

Sort

Notification 1

Notification 2

Notification 1

Notification 1

Notification 1

Done

Reset Password
Heiton

Icon

Login

Welcome to site dashboard

Reset Password


Enter New Password:






Confirm New Password:

Done

User Profile

Thao My

**PROFILE**
ADMIN JD



First Name

Last Name

Address

City


State/Zip Code


Email


Edit


Home Page


Thao My


 CreditWatch


 Admin JD


 Dashboard



 Customers

 Upload Data


 Payments

 Notifications

 Settings


 profile  logout

Home Page

	Risk Alerts 	
Risk Score	Fraud Report	
Live detections with high risks		

Create User Profile

Hannah



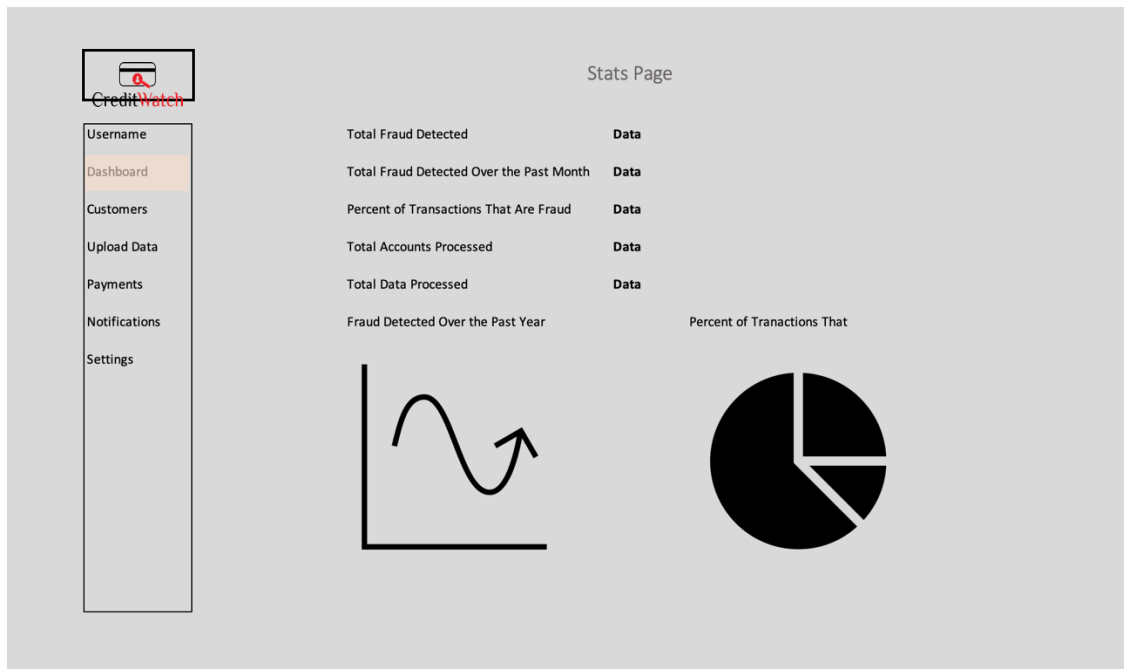
Enter User Info

First Name	<input type="text"/>
Last Name	<input type="text"/>
Password	<input type="password"/>
Verify Password	<input type="password"/>
Email	<input type="text"/>
Phone Number	<input type="text"/>
Security Question 1	<input type="text"/>
Answer	<input type="text"/>
Security Question 2	<input type="text"/>
Answer	<input type="text"/>

Create User

Dashboard

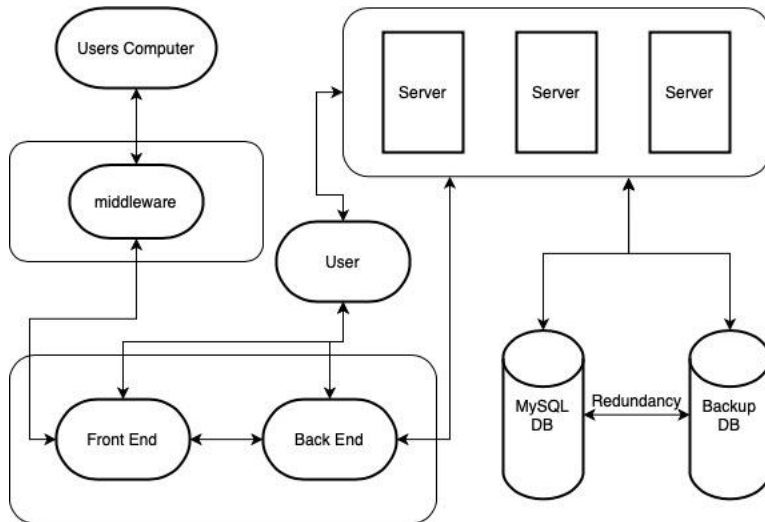
Hannah



Architecture Design

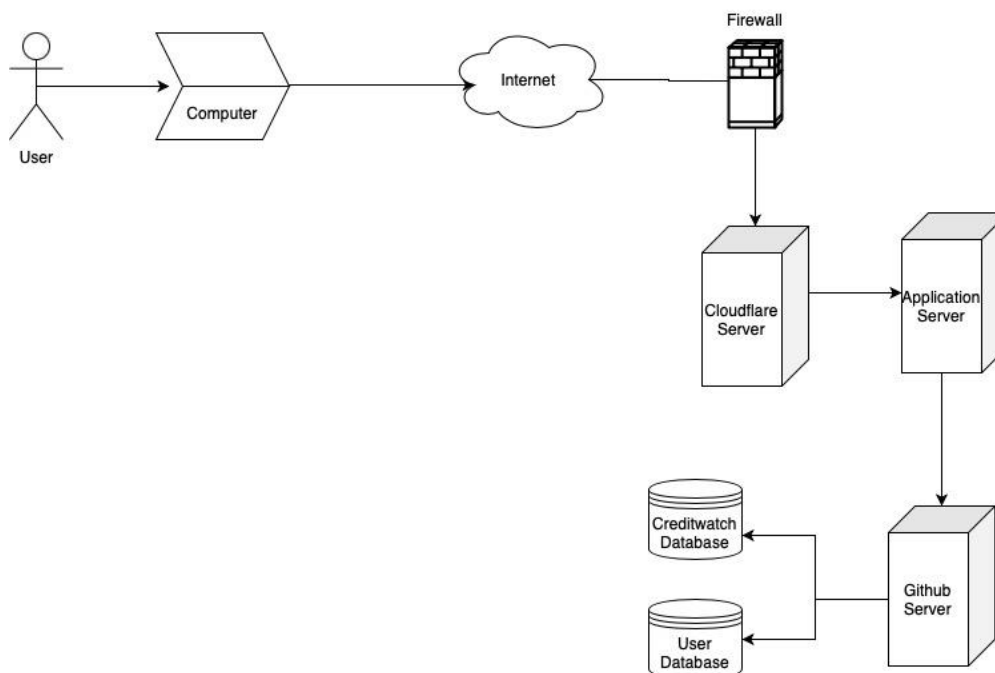
Logical Architecture

David and Thao My



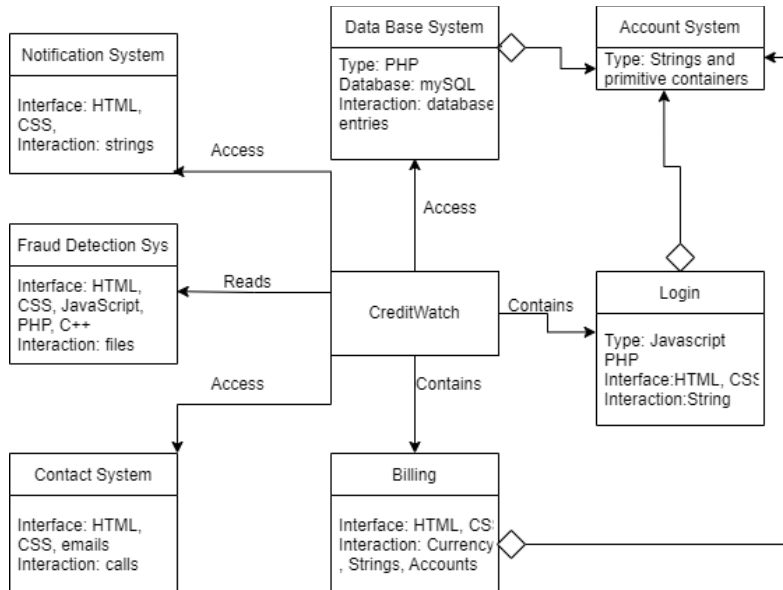
Physical Architecture

Hannah and Justin



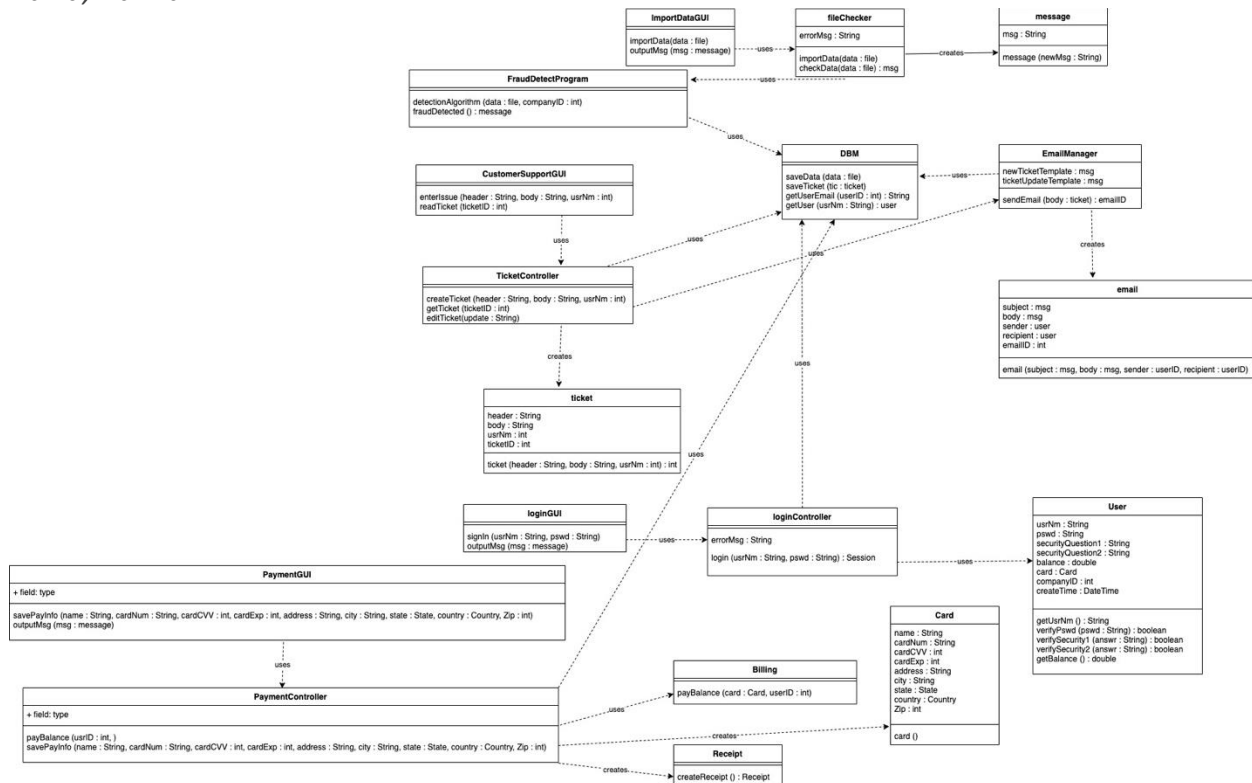
Context Diagram

Heiton and Muhammed



UML Classes

David, Hannah



Implementation

By using the agile method, more specifically scrum, we will be able to effectively create our project. Our team will reference the main functional requirements (R.1-10) to recognize our general objectives for our project. By focusing on the main functional requirements, which will act similar to a product backlog, the team is able to create sprint cycles focusing on the relevant secondary functional requirements according to their main requirement. We will incrementally create and test each individual sub function. Upon finishing all the sub requirements (Example R.1.1-R.1.12), the main functional requirement should be completed simultaneously and will then be incorporated into our website/product. After all the main requirements are finished our team will then move on to the closure phase to discuss and ensure that all parts of the product are functional to standards.

Over the process of our project, we are using Microsoft Teams and GitHub as management tools. Our code uses the C, PHP, HTML, CSS, and JavaScript languages. In addition, we are using Draw.io, VisualStudio, TextEditor, HostGator, and Microsoft Office to create our deliverables.

Software Testing

The software testing methods picked to make our application are to make sure it works as intended in the end-product. The process will go in order as they are listed starting from top to bottom. Starting with unit testing into regression testing we can assure that our program is functional from the bottom up. We will finish checking our functional testing with system testing and acceptance testing to make sure everything works as intended and all requirements are fulfilled.

Functional testing:

- Unit Testing
- Regression Testing
- System Testing
- Acceptance Testing

The Non-functional testing will consist of performance and usability testing to ensure the quality of our application.

Non-functional:

- Performance testing
- Usability Testing

Test Cases:

Number	Name	Type	Execution	Acceptance Criteria	Name
F1.1	Create User Account	Functional	From the login menu, select create account, fill in relevant info like name, email, and password and click confirm.	Email, name and password are valid, and account is created and saved in the DB	Heiton
F1.2	Edit user	Functional	From the side menu, select the user account page, and select "Edit Profile". Update user contact details and click "Save". Reload user account page.	User account changes applied are successfully updated upon revisiting the user accounts page.	David
F1.3	Specify User Access	Functional	During account registration user will input a code that is used to specify their user access.	User Access Code, a String of 14 characters	Hannah
F1.4	Change Security Questions	Functional	Login > Account Page > Password > Change Security Questions	The user successfully changes their security questions from a selected list of prompts after logging in again	Justin

F1.5	Save Billing Info	Functional	After entering payment info, prompt to save billing info will appear	User accepts saving of payment and the info is stored in the DBM	Heiton
F1.6	Change billing info	Functional	User account>view profile>edit profile> edit billing info>input new info and save		Thao My
F1.7	Specify user works for	Functional	User account>view profile>edit profile> edit company> input company and save	System checks if user registered under company and saved to DBM	Muhammad
F1.8	Save user contact info	Functional	If no contact info is saved in the system for a user's account then when user input new contact info in their account then save that info	Correct form of US addresses and phone number being processed and save to DB	Thao My
F1.9	Verify User contact Info	Functional	From the side menu, select the user accounts page. Select view contact info	User is prompted to reenter password and then contact info is then shown	Heiton

F1.10	Edit Username	Functional	From the side menu, select the user accounts page. Select edit username	User is prompted to edit new username and then screen is set back to user accounts page.	Hannah
F1.11	Edit Password	Functional	From the side menu, select the user accounts page. Select edit password	User is prompted to edit new password and then the screen is set back to user accounts page	Hannah
F1.12	Verify password meets rules	Functional	From the side menu, select the user accounts page, and select "Update Password". Input a password that does not meet the password complexity requirements and click "Save".	User is prompted upon trying to save an insecure password, not meeting the configured complexity requirements , with a popup message stating "Password does not meet the requirements , please try another.".	David
F1.13	Delete User Account	Functional	Login > Account> Delete Account > Confirm Delete	The user's account no longer exists within CreditWatch's database.	Justin

F2.0	Generate temporary password	Functional	From the side menu, select the user account page. Generate a new account, or request a password reset.	A temporary password is generated and emailed to the user within 3 minutes.	David
F2.1	Send Temporary Password with Account Creation	Functional	Creation triggers a temporary password to be created, which in turn sends an email with the password to the user.	User uses the new password to log in.	Hannah
F2.2	Deactivate password after set amount of time	Functional	From the side menu, select the user account page. Click "Register" to create a new account. Do not login to the new account.	After 7 days, an email is sent notifying the user that the temporary password has expired, and a password reset is required to access the account.	David

F2.3	When signing in with temporary password ask for new password	Functional	Home page > log in with temporary password > prompt to create new password	User enters a new password that fit all criteria such as a password must have at least 1 upper case letter, 1 numeric number, special characters and lower case letters	Thao My
F3.1	Verify password and username combo	Functional	On the side menu, select the users account page or another protected page. Once greeted with the login page, login with invalid credentials.	Login should fail immediately and provide GUI message that login was unsuccessful	David
F3.2	Reset Password	Functional	User goes to login page and selects reset password. User fills in relevant info and is prompted for a valid password to reset to.	The password is valid and the reset process, like security questions, are passed	Heiton

F3.3	Security Questions	Functional	Create Account > Login > New Password > Create Security Questions	The user picks from the list of security question prompts and inputs their own answers into two of them.	Justin
F4.1	Track Customer Issues	Functional	Customer Support will update ticket in system whenever process in solving customer issue moves forward.	Customer issue is still active and communication is being held between Customer Support and User.	Muhammad
F4.2	Respond to Customer issues	Functional	Customer Support will respond to customer issue through system	User sends issue that requires Customer Support aid.	Muhammad
F5.1	Send Receipts to customers	Functional	The system will send the customer a receipt after having paid for CreditWatch's services	The user successfully gets sent a receipt.	Justin
F5.2	Tracks payments made and due	Functional	System will record transactions made and if they are successfully paid or need to be paid	User makes a transaction and pays now or chooses to pay later.	Muhammad

F5.3	Charge for services according to time and amount of data	Functional	System will record amount of time and data user used from services and will charge user accordingly on payments tab	User uses the services provided by CreditWatch for any amount of time.	Muhammad
F5.4	Stop services when bills are overdue	Functional	From the side menu, select the user account page. Click "Register" to create a new account and do not pay the payment due for services.	Account services are deactivated immediately after the payment due date if no payment is received.	David
F6.1	Import Data	Functional	Account (customer) > Upload Data (section) > Upload Data (action)	The data follows the specified criteria and formatting to be analyzed	Justin
F6.2	Sort Data	Functional	System will take in a list of data and sort it with an algorithm	The list of data is valid and accessible	Heiton

F6.3	Search data for potential fraud	Functional	System will use an algorithm to create a trend in the data recordings of transactions made. System will use this trend to search for any out of the ordinary transactions made on the user's credit card.	Data is in correct format and includes all the needed information.	Muhammad
F6.4	Learn From Past Data	Functional	The system will keep records of successful fraud detection to detect similar patterns in the future	The system Successfully Keeps track of past successfully detections	Justin
F7	System will protect data	Functional	All data sent is protected by not saving data on the website and in plaintext	Data is protected if non users don't see protected data in everyday use.	Heiton
F8	System shall send fraud detection	Functional	User is contacted using saved contact information about possible fraud	User contact information is working and valid	Heiton

F9	Send notifications about maintenance	Functional	Visit the admin panel and from the side menu, select maintenance. Select schedule a maintenance window. Provide time and date of the expected maintenance, select the length of time to be scheduled, and submit the maintenance window form.	Banner's displayed advertising the scheduled maintenance 7 days prior on all login pages. Emails are sent providing the same message, one 7 days prior and again 1 day prior.	David
F10.1	Track Total Company users	Functional	Login (employee) > Admin panel > Side menu > statistics	The user sees the number of users that have been created	Justin
F10.2	Track total accounts being processed	Functional	Login (employee) > Admin panel > Side menu > statistics	Correct number of accounts being processed is calculated and displayed	Thao My
F10.3	Calculate total fraud detected	Functional	Login (employee) > Admin panel > Side menu > statistics	Correct number of fraud detected being processed is calculated and displayed	Thao My

F10.4	Calculate percentage of users total at risk	Functional	Login (employee) > Admin panel > Side menu > statistics	Correct number of percentage of users being processed is calculated and displayed	Thao My
F10.5	Calculate percentage of users at risk in specific company	Functional	Login (employee) > Admin panel > Side menu > statistics	Correct number of percentage of users being processed is calculated and displayed	Thao My
NF1	Process Quickly	Non-Functional	System will be programmed to have pre-processed data trend so that once a transaction occurs that does not follow the pattern and is out of that data trend, The system will catch that instantly and notify the user.	As soon a transaction occurs, system will instantly notify user if transaction doesn't follow trend.	Muhammad

NF2	Process in Real Time	Non-Functional	System will check each transaction whenever taken place for any fraud based off previous data and trends and will notify user instantly.	User will receive notification of potential fraud in a matter of seconds after a transaction takes place on user card.	Muhammad
NF5	Store Data for 3 Years	Non-Functional	Data older than three years is deleted, otherwise it is saved in our database	Data older than three years is deleted	Justin
NF10	Welcome Page	Non-Functional	After users logged in welcome screen will display	user's account is validated when logging in and a welcome page will shown	Thao My

Executed Test Cases:

Test Case:	Inputs:	Expected Output:	Actual Output:
Create User Account	Username, Password, Password confirmation, Email, security question, security question answer	Create account and new session	Create account and new session
Edit User	First name, last name, phone number, address 1, address 2, city, state, zip, email, country	update user profile	update user profile
Edit password	new password, confirm new password, security question answer	Update Password	Update password
Verify Password Meets Rules	Suggested password	Return true	Return true
Verify Password and username combo (login)	Username, Password	New session	New session
Welcome Page	Input, Successful login	User sees the welcome page	User sees the welcome page
Security Questions	Security question answer, security question	security question created	security question created
Charge for services according to time and amount of data	number of activations, number of customers, price of product	Bill	Bill
Import data	CSV Data file	Fraud results	Fraud results
Sort Data	Data	Data sorted by customer ID, then time	Data sorted by customer ID, then time
Export Data	Export Request	CSV file	CSV file
Search data for potential fraud	Data	Fraud Detected Data	Fraud Detected Data
Learn from past data	Transaction Data	Averages of transaction fields	Averages of transaction fields

System will protect data	Password	Encrypted password	Encrypted password
Track Total company Accounts	CSV file – data from database	Count of unique accounts	Count of unique accounts
Track total users being processed	Data from database	Count of unique users	Count of unique users
Calculate total fraud detected	Data from database	Count of fraud items	Count of fraud items
Calculate percentage of users total at risk	Count of total users, Count of fraud items	Percentage of users at risk	Percentage of users at risk
Process quickly	Data	< min processing time	< min processing time
Process in real time	Data	Fraud results within 1 minute	Fraud results within 1 minute

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38			
Calculate percentage of users at risk specific company at risk																																									
Calculate percentage of users total at risk																																									
Track total fraud detected																																									
Track total accounts being processed																																									
Track total company users																																									
CreditWatch shall send notifications about maintenance																																									
CreditWatch shall send fraud detection notifications																																									
CreditWatch shall protect data																																									
Learn from past data																																									
Search data for potential fraud																																									
Sort data																																									
Import data																																									
Stop services when bills are overdue																																									
Charge for services according to time and amount of debt																																									
Track payments made and due																																									
Send receipts to customers																																									
Respond to customer issues																																									
Track customer issues																																									
Use security questions																																									
Reset password																																									
Verify password and username combo																																									
Create new password when using temporary password																																									

Deployment

Initial Deployment Procedure

1. Set up the CreditWatch and MySQL database and that servers are running. The database will allow our customers to log in and access all the data relevant to them and to upload any additional data.
2. The database must be filled with data from CSV files to be analyzed and put into the CreditWatch database. Additionally, the server and database will synchronously work to produce results showing total fraud, monthly, fraud, percentage of fraud, transactions checked, and their sources on their dashboard.
3. Ensure that the database is getting all the information from the server and all data being returned and received are functioning as expected.
4. Through our website, allow users to create their accounts providing a username, password, email, and security question info.

Subsequent Deployment Procedure

1. On Saturday at 3AM CST, make any necessary updates/changes to our code based off user feedback and observed results from our database.
2. With minimal downtime to our website, update the code and make necessary adjustments to our CreditWatch database.
3. Monitor that the website, server, and databases are connected and ensure they are running correctly after an update is rolled out.

Glossary of Terms

ⁱ CreditWatch	Software designed to detect credit card fraud
ⁱⁱ Fraud	An instance where a credit card is used by a nonauthorized person
ⁱⁱⁱ Customers	Companies that issue credit cards
^{iv} User Accounts	Accounts held by our company for individuals to access their data
^v User Access	The access a user account has to data. For example the access will be specific to the company a user works at
^{vi} Services	Access to use CreditWatch to detect credit card fraud
^{vii} Maintenance	CreditWatch will be unavailable for a certain amount of time for updates
^{viii} Logistics	Calculated data about CreditWatch user accounts and fraud detected
^{ix} Detected	CreditWatch has calculated that a transaction is likely fraud