AVOCADO TOAST: AN ONLINE BANKING PLATFORM

DAMH PHAM ANANDITA DUBEY FLAVIU TAMAS
CARLOS DELEON ALEX PETROS

CONTENTS

LIST OF FIGURES

1 GROUP INFORMATION

Project name	Banking application
Class	Software engineering
Semester	Spring 2019
Group	1
Group members	Damh Pham, Anandita Dubey, Flaviu Tamas (coordinator), Carlos Deleon, Alex Petros
Submission date	2019–02–01

Table 1: Group information

2 RESUMES

2.1 Damh Pham

2.1.1 *Classes taken*

- Data structures
- System-level programming
- Math models

2.1.2 *Technologies*

- Languages: Java, C, Python

2.1.3 *Experience*

None

2.2 Anandita Dubey

2.2.1 *Classes taken*

- Data structures
- Computational intelligence
- Mobile app development
- Intro to machine learning
- Operating systems

2.2.2 *Technologies*

- Languages: Java
- Skills: Android Studio, Neural Networks

2.2.3 *Experience*

None

2.3 Flaviu Tamas

2.3.1 *Classes taken*

- Data structures
- Computer organization
- System-level programming
- Programming language concepts
- Computer architecture
- Parallel & distributed programming

2.3.2 *Technologies*

- Languages: JavaScript, HTML, Java, Scala, CSS, C, Bash, SQL, Nim
- Libraries: Backbone.js, Underscore.js, React, Django, PCRE
- Other: Git, PTC Integrity, RST, SOAP, LATEX, Bugzilla, Wercker, Gitlab CI, Docker

2.3.3 *Experience*

At my previous job I worked on doing web development on embedded systems, and worked on basically everything, from the front-end code to the analysis code & system admin stuff for the Geopyc 1365. I worked in Python on the backend, and Javascript/Typescript/Backbone/React on the frontend.

My current job is doing devops, as well as a bit of embedded programming, for GTRI. I've mostly been using Python, as well as a decent amount of bash & C++.

2.4 Carlos Deleon

2.4.1 *Courses Taken*

- Data structures
- Computer architecture
- System-level programming
- Programming language concepts

2.4.2 *Technologies*

- Languages: Java, C, Assembly, HTML, CSS

2.4.3 *Experience*

None

2.5 Alex Petros

2.5.1 *Classes taken*

- Data structures
- Computer organizations
- System-level programming
- Windowing systems

2.5.2 *Technologies*

- Languages: Java, C, Bash, SQL

2.5.3 *Experience*

None

3 WORK BREAKDOWN

Assignee	Username	Task	Duration (h)	Depends on	Due date
Anandita	demo123git	User requirements (1–5)	2	Github, GroupMe	2019–01–31
Alex	apetros1	System architecture	2	Github, GroupMe	2019–01–31
Carlos	cdele	Teamwork basics	2	Github, GroupMe	2019–01–31
Danh	nessico	User requirements (6–10)	2	Github, GroupMe, System architecture	2019–01–31
Flaviu	flaviut	Create Github & GroupMe; Copy-edit report	2	None; other sections	2019–01–26; 2019–01–01

Table 2: Work breakdown

4 TEAMWORK BASICS

4.1 Ground rules

Setting some ground rules can help clearly lay out what to and what not to do when working on assignments or communicating with the team.

1. **Work norms**: Who will do what and when is it due? Will anyone review it?
   - These rules are the basis for completing any group assignment and must be made clear because not making these clear can make some members think the deadline is a different date or two members could do the same part.
2. **Facilitators Norms**: Will you use facilitators?
   - Facilitators are people who will keep the team on track. They are the person who will chime in "Ok so what we are doing is…" clarifying the objective or "Guys if we finish this part now, we can go home early" motivating the group to stay on track.

     I had a groupmate in my system-level class that was like that. Having a person clarifying what we did the last meet and what we needed to do during this one was beyond helpful.

     We were working on a library checkout program and sometimes during the meetings the group, myself included, would over focus on one small part of the assignment, like the book properties. She would always get the team to focus on the requirements, and asked people in the group

who weren't enthusiastic about the project for their input to keep them engaged.

3. **Communication Norms**: How will the group communicate? Email, GroupMe, etc.?
    - Having everyone in a group chat can be helpful to facilitate communication. It can also help pass along the assignments to the leader.
4. **Meeting Norms**: When is a good time to meet and where? What if someone misses a lot of meetings?
    - These can help provide structure because in previous groups when one person is not going to meetings you can tell them that the times are clearly laid out. Without it they probably felt no consequence because there was no clarification in the beginning that not going would be met with telling the professor or asking for a grade reduction.
5. **Consideration Norms**: Can people eat at meetings or smoke? Is everyone comfortable with the other team members?
    - It's important to make sure everyone is comfortable, and it can help those teammates who are shy and may only provide input if the environment is not intimidating.

4.2 Goals

It's important to make sure that the whole team is on the same page with regards to the goal. If one teammate is only concerned with passing and getting a C or above and other members are working for an A, there will be conflict which should have been addressed from the beginning.

4.3 Hints for handling difficult behavior

If a teammate is too talkative, it can be due to their eagerness or they want to show off their knowledge to the group. If they can't take a hint that other people would like to provide input, then you can use humor to stop them from dominating the situation, or alternatively you can talk to them privately about letting others provide an equal amount of input.

If they are too quiet then you can make an effort to ask them questions about themselves and tell them you appreciate any input they give. It is likely that they have good ideas for the project because they have been listening the whole time.

If the person is very aggressive and critical then you should evaluate their critiques and then ask them privately to tone down their actions, as they are having a negative effect on the rest of the team.

If a teammate complains a lot, evaluate the complaints and make changes to resolve those complaints. If it is not possible to do so, then talk to them to try to resolve their problem in another way.

4.4 Hints for handling group problems

The whole group may come across problems that affect all members.

If the team is floundering and not really getting anything accomplished getting a list of tasks can be very helpful. In my experience, someone taking up the leadership role can help provide the group with needed direction.

When the team is constantly getting distracted and going on tangents you can ask for an explanation of a certain part of the project in order to discretely get the group on topic.

If the group makes decisions too quickly then you can ask things like "Are we all in agreement about this?" to slow things down, and provide more time to find an optimal solution.

If the group fails to come to a consensus, then multivoting may be a way to finally make a decision. Multivoting is listing the choices and having members rate them from best to worst and find similarities in the best ideas and try to combine them.

If team members are fighting between themselves, then take a break and use the techniques from Hints for Handling Difficult behavior.

If someone is being ignored or ridiculed then they will not be able to provide the team with input. Therefore, all members must make a extra effort to be inclusive and make everyone feel accepted.

When a group member doesn't want to do their share of work, then it's necessary to talk to them and list the consequences of their actions. Hinting at this is insufficient, it's important for things to be fully explained to ensure they fully understand. Not being direct could get them to not like you and resist doing the work even more.

5 PROBLEM STATEMENT

5.1 What is your product, on a high level?

Our product is an online banking application, which will allow the customers of a bank or other another financial institution to perform money-management without physically visiting the bank.

It will allow the customers to open their accounts, manage them electronically, to monitor them, to make transactions, pay their bills, transfer money, make deposits, and so on.

5.2 Whom is it for?

An online banking application is beneficial to everyone but is especially useful for those people with a stringent work schedule. It will help them to manage their accounts and keep track of their activities in a quick manner with minimal costs without the need to to visit a physical bank during working hours or make a phone call.

5.3 What problem does it solve?

- 24/7 availability, saving the customer from rushing the banks during working hours. With an online banking system, people can perform their tasks at a time that suits their work schedule.
- Stringent schedules, where customers with strict working hours to perform their banking activities effectively and conveniently.
- Centralized source of information, where instead of visiting different officials specialized in different tasks, the customer can use an online banking application flexible enough to do any task in a single click. Human bankers are not always available, but the application always will be, meaning there is no need to rush to the bank to get things done.
- Remain informed: With the online banking system, customers can easily receive up-to-date information regarding their upcoming deadlines or dues through notifications, emails, or text messages.

- Easy bill payments, so that there is no need to rush to the bank. Everything can be done at home instead.

5.4 What alternatives are available?

- Services provided by 3rd party application, like Mint
- A phone app
- In person banking services
- Phone calls
- ATMs

5.5 Why is this project compelling and worth developing?

- It would reduce costs for banks by reducing the amount of human labor required.
- An electronic banking system would enable banks to keep stringent records
- A computerized ledger would reduce the number of mistakes when calculating interests, making transactions, and so on.
- It would increase customer satisfaction because they would have access to our banks from any place with WiFi.

5.6 Describe the top-level objectives, differentiators, target customers, and scope of your product.

- Objective: To create a product that performs to our requirements and have it sustainable for multiple lifecycles.
- Differentiators: Our system won't sell data to any third parties, have no hidden transaction fees, and will not participate in predatory lending practices
- Target customers: Our target customer is the average citizen, anyone that currently uses or will use a bank for their transaction needs.
- Scope: The scope of our project is building an online business ledger, making a database to store our information, developing a web app and making a UI for our customers.

5.7 What are the competitors and what is novel in your approach?

Our competitors are Finacle, nCino, Oracle, etc. What we bring new to the table is that our system will be entirely online. Additionally, our platform is customer focused and will be fully transparent. We won't sell customer data or charge hidden fees.

5.8 Make it clear that the system can be built, making good use of the available resources and technology.

Our system is possible because it will be very simplistic and direct. Clients will connect to our application software where they will be greeted by a user-friendly interface, which will primarily be coded in HTML, with CSS used to create a beautiful design for our clients. Finally, the frontend of our application will be connected to a backend SQL database. The database will be responsible for recording transactions such as withdrawals, deposits, bill pays, and more.

5.9 What is interesting about this project from a technical point of view?

The project will use a client-server architecture in order to be accessible to clients from anywhere.

Emphasis will also be placed on graphical design, using our creativity to design a nice interface for our clients.

6 SYSTEM REQUIREMENTS

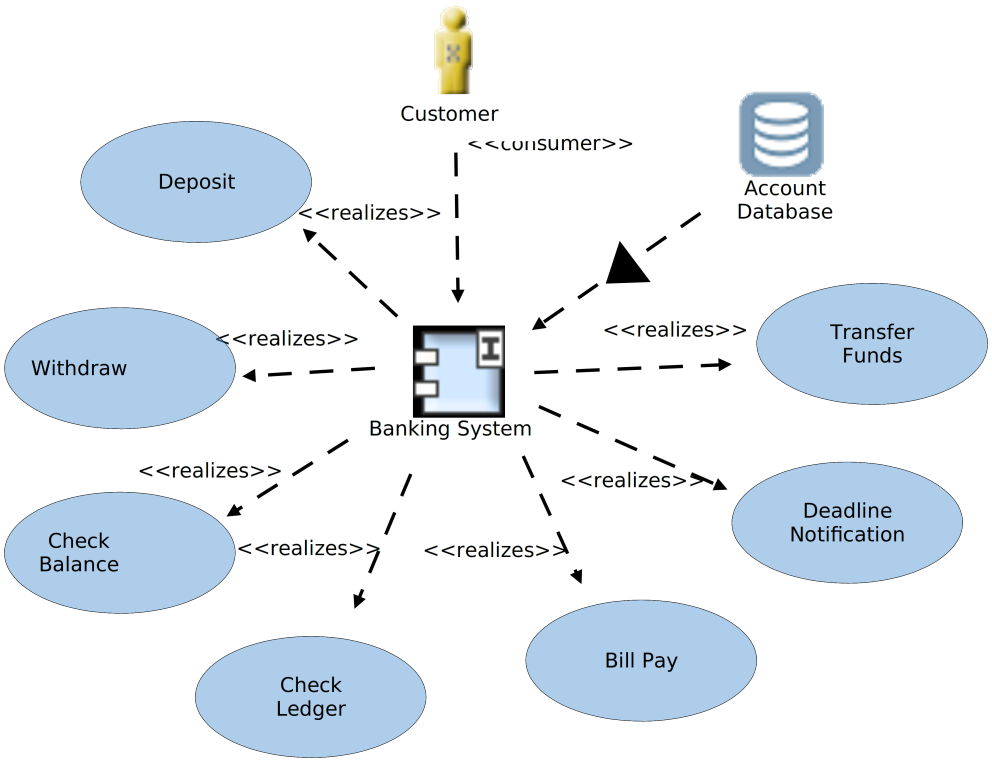

Figure 1: Context model diagram of the system

A SCREENSHOTS

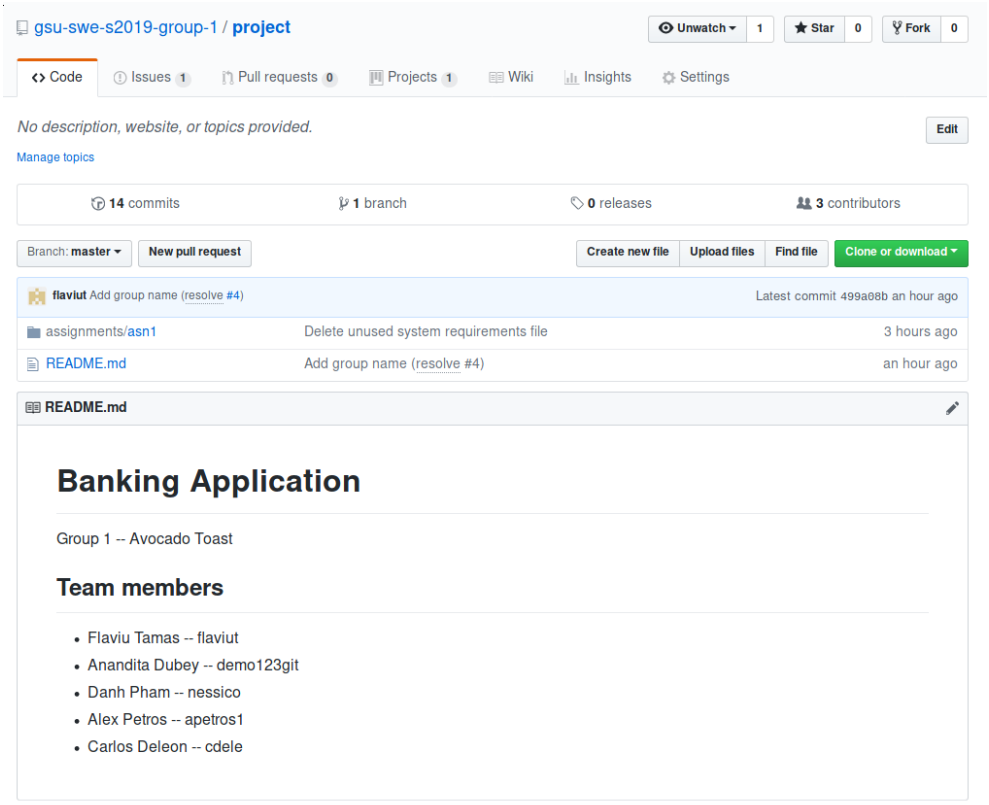

Figure 2: Screenshot of our group's GitHub readme

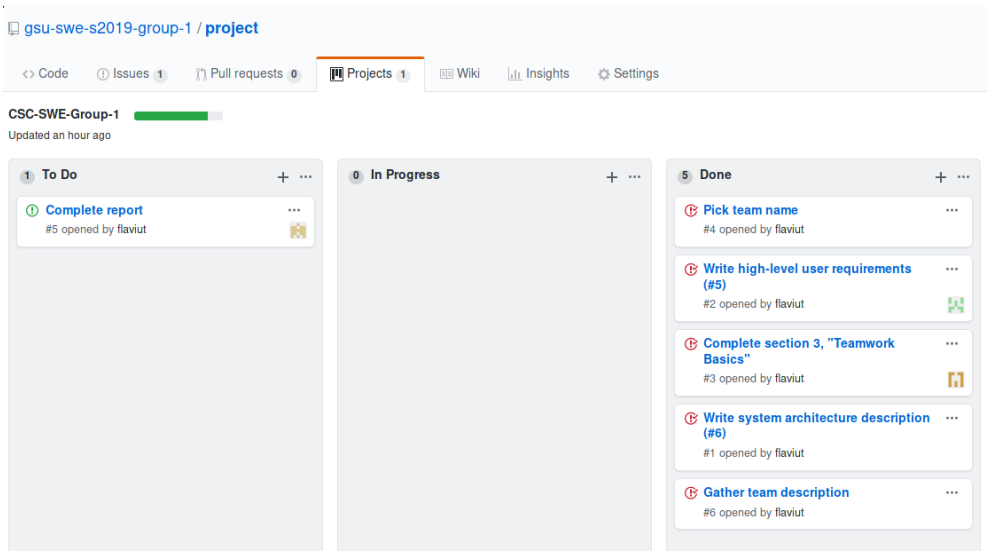

Figure 3: Screenshot of our group's Kanban board