Role - Solution Architect

The solution architect is responsible for the technical direction of the project, making decisions about the strategy to be used to fulfill the requirements, both functional (business functionality) and nonfunctional requirements (speed, failover, etc.) Implementation technologies, platforms and tools and coordinating this with the developers and testers.

Revision History

Week Number	Author	Description of changes
1-2 (Turn-in 1)	Victoria Mannina, David Chan	Chose Database (Parse.com), Server (Parse.com), Language (Java), Platform (Android App)
3-4 (Turn-in 2)	Victoria Mannina, David Chan	Created UML Diagram, Chose Version Control (Github) and Android IDE (Android Eclipse SDK), development continuing with decisions from Turn-in 1 (above)
5-7 (Turn-in 3)	Matthew Nguyen, Michael Tran	Revised UML Diagrams to reflect refactoring, including Sequence Diagram for credit and included Design Pattern Description

1. Architectural Decisions

Options

What are the technology options available that you considered? List them here.

Decision

You have to go with one of the options. Describe the details of that option here.

Rationale

Why did you make this choice? Describe the detailed rationale, the reasons for making the choice at this time in your project. You may need to change this decision later, but note that the change will have repercussions such as code rewrite etc.

NOTE: All decisions are listed, scroll down for newest architectural decisions.

Architectural Decision Number	1. Chose Database
Options	Parse.com, Mysql

Selected Option	Parse.com
Rationale	We found this and decided it fit what we needed perfectly. It had all the capabilities and is free for a limited amount of requests per second. We didn't feel there was a need to keep searching because we didn't need anything else as far as capabilities and it couldn't be any cheaper.
Architect	Victoria Mannina and David Chan
Reviewed and Signed off with the following people at this date/time	Austin Han, Matthew Nguyen, Jason Dizon, John Chan, Mike Griffin, Michael Tran, Chesong Lee, 10/18 10:00 AM

Architectural Decision Number	2. Chose Server
Options	Parse.com, Google Cloud Platform
Selected Option	Parse.com
Rationale	We felt this was best since we could also use it for our Database. Chesong has had previous experience with Google Cloud Platform and said there is a learning curve. We want something quick to learn and easy to use.
Architect	Victoria Mannina and David Chan
Reviewed and Signed off with the following people at this date/time	Austin Han, Matthew Nguyen, Jason Dizon, John Chan, Mike Griffin, Michael Tran, Chesong Lee, 10/18 10:00 AM

Architectural Decision Number	3. Choose Language
Options	Java, C, C++, PHP, Objective-C, HTML
Selected Option	Java
Rationale	We wanted to use something we all have experience with. We also feel this is the easiest to use for Android development.
Architect	Victoria Mannina and David Chan
Reviewed and Signed off with the following people at this date/time	Austin Han, Matthew Nguyen, Jason Dizon, John Chan, Mike Griffin, Michael Tran, Chesong Lee, 10/18 10:00 AM

Architectural Decision Number	4. Choose Platform
Options	Android, iOS, Website
Selected Option	Android
Rationale	We all wanted experience with Andriod development. We all know Java, also, so we wanted a platform we could use Java on (not iOS).
Architect	Victoria Mannina and David Chan
Reviewed and Signed off with the following people at this date/time	Austin Han, Matthew Nguyen, Jason Dizon, John Chan, Mike Griffin, Michael Tran, Chesong Lee, 10/18 10:00 AM

Architectural Decision Number	5. Choose version control
Options	Bazaar, Github, LibreSource, GNU arch
Selected Option	Github
Rationale	The team wanted a protected, solid, easy to use version control program. Although there are many types of version control programs, the team decided on Github since it is a trusted, easy to use, and well known version control.
Architect	Victoria Mannina and David Chan
Reviewed and Signed off with the following people at this date/time	Austin Han, Matthew Nguyen, Jason Dizon, John Chan, Mike Griffin, Michael Tran, Chesong Lee, 11/1 10:00 AM

Architectural Decision Number	6. Choose one Android Integrated Development Environment
Options	Eclipse with SDK, Dr. Java
Selected Option	Eclipse SDK
Rationale	Eclipse is a very well known java development environment and is the most known in compatibility with android SDK.
Architect	Victoria Mannina and David Chan
Reviewed and Signed off with the following people at this date/time	Austin Han, Matthew Nguyen, Jason Dizon, John Chan, Mike Griffin, Michael Tran, Chesong Lee, 11/1 10:00 AM

2. Architectural Diagrams

1.1. Logical Architecture

[redacted]

[redacted]

1.2. Physical Architecture

Comment:

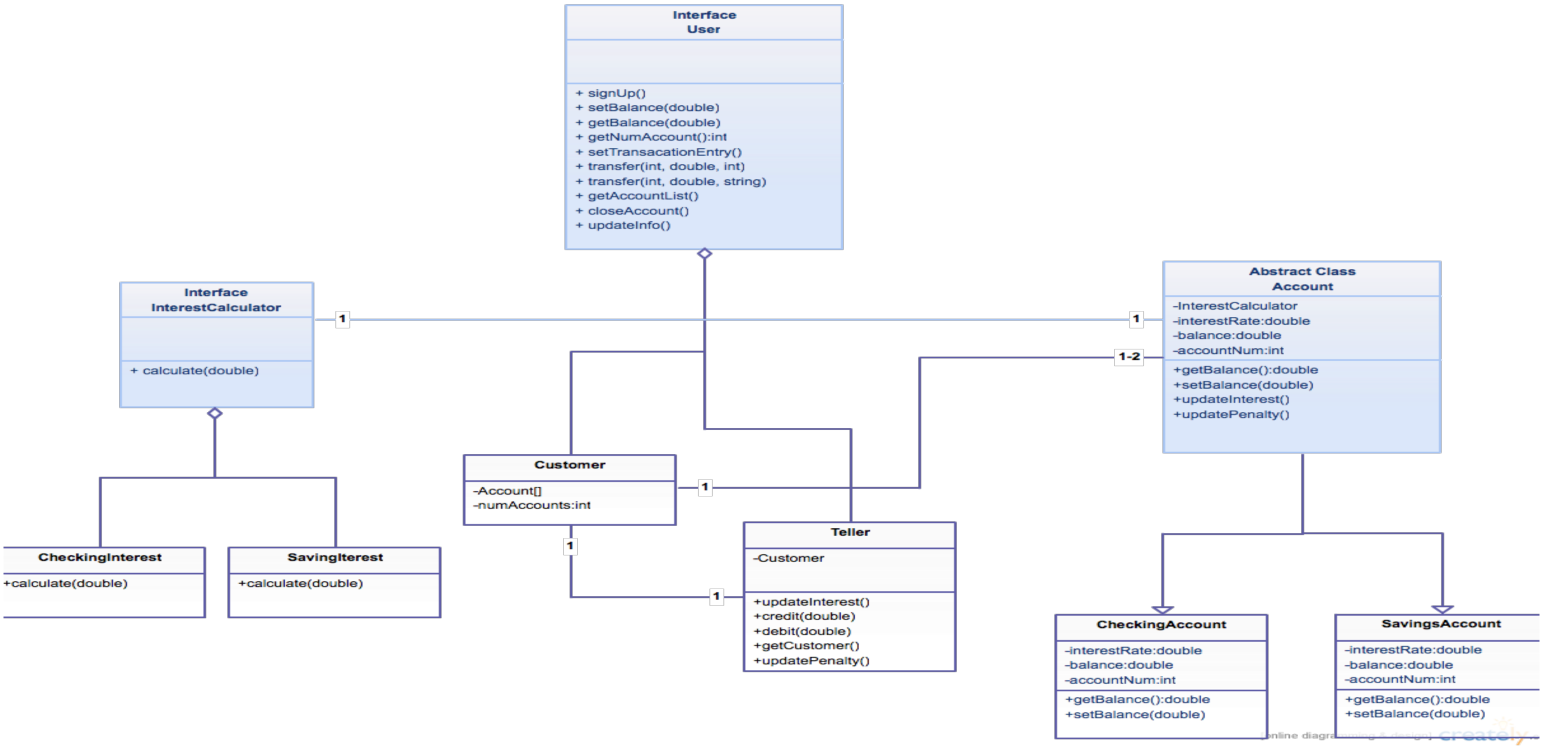
- *How are the components really connected?*

The physical servers of Parse.com will be hooked up to the internet and accessible to our mobile app. Meanwhile, our mobile app will be communicating to those servers with the mobile device's connection to the internet via1 port 80, 443, or other Parse.com available ports.

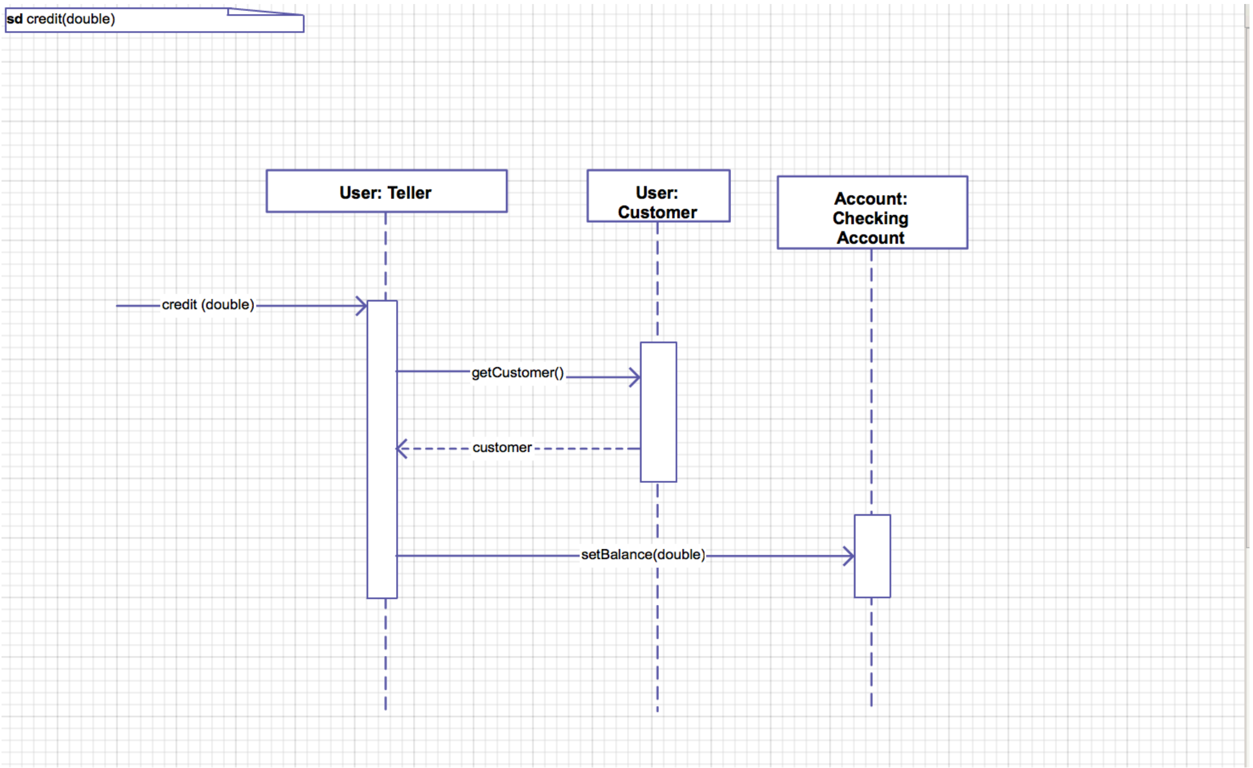
With sensitive information, our mobile app will be using the HTTPS protocol to assure a secure communication path between our user's device and our servers.

Our application software will be running on the servers of Parse.com, listening to any authorized request of services from our mobile device application. When a mobile user interacts with our application, the mobile device will send a request to our servers with an authentication encryption. Provided that the user had input the correct credentials, the inquire authorization is confirmed through the designated listening port of our server software. Our server will then process the request and securely respond with the appropriate data or change confirmation to the correct mobile device port.

UML Diagram



Sequence Diagram



Design Patterns

1. Strategy Pattern

   We use the strategy pattern when there are different ways to implement something across different account types such as checking accounts and savings accounts. There are different interest rates for checking accounts and savings accounts. We used strategy pattern in the InterestCalculator interface which is implemented by CheckingInterest and SavingInterest classes which have different algorithms for calculating the interest due to the different interest rates. The context is the Account class, the strategy is the InterestCalculator, and the concrete strategies are the SavingInterest and CheckingInterest.

2. MVC Pattern

   We use the MVC pattern in the overall implementation of our Android application. The Model is implemented by our Parse back-end and we query Parse for any data. The View is implemented by all of our Activity views and the XML files. The Controller is implemented largely by our User interface and there are listeners that listen for user inputs and changes to the Model and View are made accordingly.

3. Factory Pattern

   We use the factory pattern with the use of the Parse objects. Knowledge of the creation of the Parse objects is hidden by the Parse database, and we use queries to get the information we need for our customer accounts such as name, balance, etc.

Iteration Review Questions

NAME: Michael Tran, Matthew Nguyen

TEAM NAME: Honey Badgers

ROLE: Architect

- Is everyone happy with the quality of work? Documentation? Testing?

Yes, everything seems to be going according to plan.

- How did everyone feel about the pace of the iteration? Was it frantic? Reasonable? Boring?

The iteration feels well faced but seems frantic since there are always constantly changing requirements.

- Is everyone comfortable with the area of the system they were working in?

I believe everyone is confortable in the area they are working in.

- Are there any tools particularly helping or hurting productivity? Are there any new tools the team should consider incorporating?

As architect, we found that Creately was extremely helpful in creating UML Diagrams and Axure was excellent for wire framing.
We have not considered any new tools at this time.

- Was the process effective? Were any reviews conducted? Were they effective? Are there any process changes to consider?

Yes the process was effective and aided our work.

- Was there any code identified that should be revisited, refactored or rewritten?

Most of our code is being refactored to match the new UML diagram and implementing a Strategy, MVC, and Factory pattern.

- Were any performance problems identified?

At this time there are no performance problems with the design.

- Were any bugs identified that must be discussed before prioritization?

There are unknown bugs that cause crash when requesting from Parse.

- Was testing effective? Is our test coverage high enough for everyone to have confidence in the system?

No, testing is done by the Architect

- Is deployment of the system under control? Is it repeatable?

Deployment is under control and repeatable.