Swift Loan Processing

Electronic Loan Application And Processing

Charles Hathaway
Hoang Phan
Emily Russell
Kyle Timins

December 3rd, 2012

What is Swift Loan Processing?

- Semi automated loan processing
- Online process
- Quicker processing
- Background processing

Who is this for?

- Fast paced society
- People who prefer to do banking online

Functional Requirements

- Allow users to register accounts
- Store redundant basic information
- Require secure authentication of users
- Allow users to apply for loans
- First version must allow users to apply for collateralize personal loans and collateralize consumer loans
- Allow users to view existing or previous applications
- Allow users to cancel applications within the first 24 hours

Functional Requirements

- All user interactions will take place in a webpage.
- The user will input data through webpage
- During a single session, the web page will keep forms populated
- To be able to do any interaction, the user must be logged in.
- All completed documents will be provided in the form of PDF documents and will be available via downloads.

Software Architecture

- Django Framework
- MySQL Database
- Two separate applications
- 4-tier Architecture
- TLS/SSL Encryption for end-user security

Software Architecture

Software Architecture

- View - what is seen by customer

Software Architecture

- View - what is seen by customer
- Web Application - Prepares forms, parses input, renders templates

Software Architecture

- View - what is seen by customer
- Web Application - Prepares forms, parses input, renders templates
- REST API - Communicates with application server

Software Architecture

- View - what is seen by customer
- Web Application - Prepares forms, parses input, renders templates
- REST API - Communicates with application server
- Data Processor - Processes applications, retrieves information

Software Architecture

- View - what is seen by customer
- Web Application - Prepares forms, parses input, renders templates
- REST API - Communicates with application server
- Data Processor - Processes applications, retrieves information
- Database - Stores local copy of customer information

Network Topology

- Servers
- Routers
- Misc.

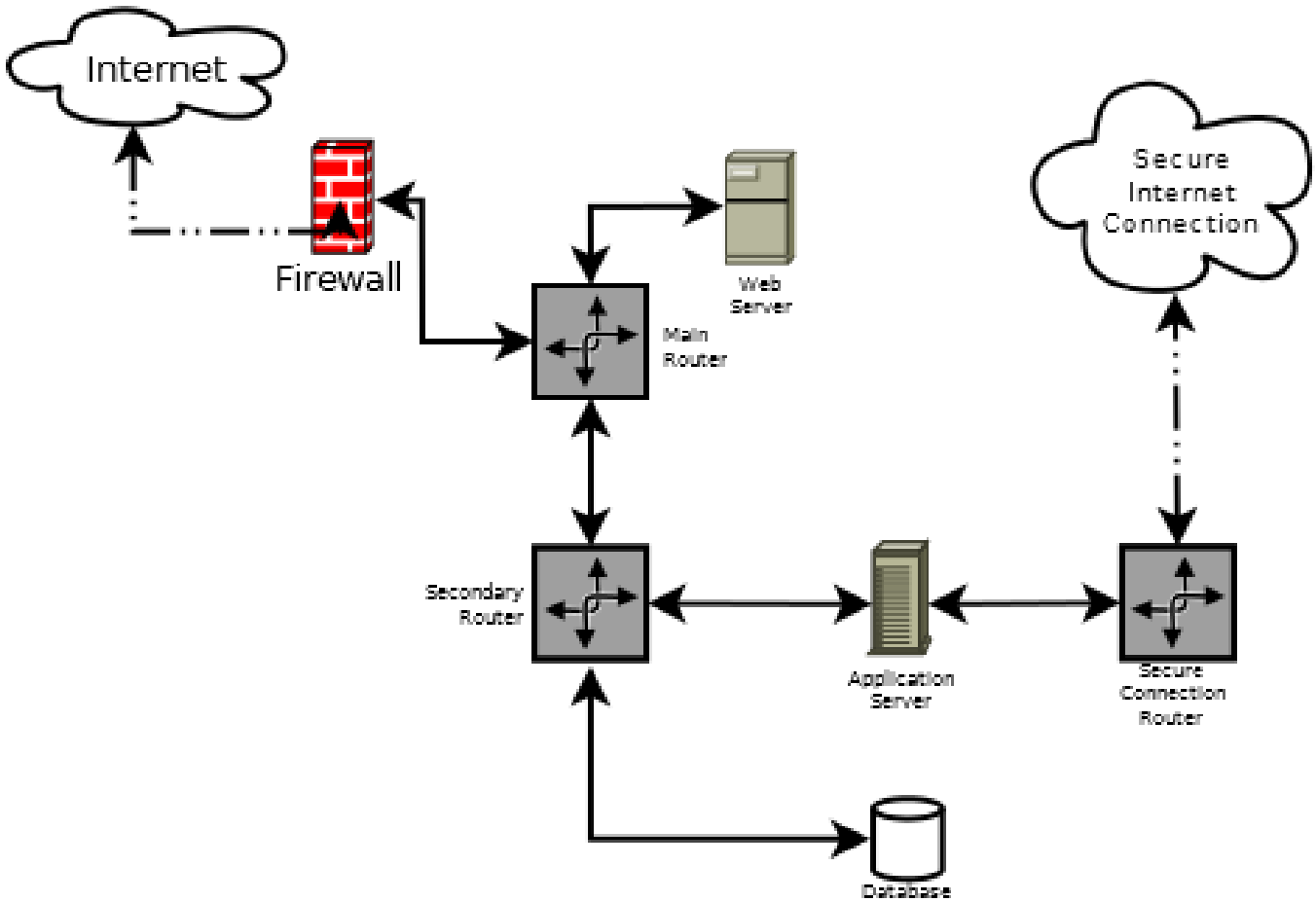
Network Topology

- Servers
  - Web Server
  - Application Server
  - Database
- Routers
- Misc.

Network Topology

- Servers
  - Web Server
  - Application Server
  - Database
- Routers
  - Main Router
  - Secondary Router
  - Secure Connection Router
- Misc.

Network Topology Cont.



Software Problems Addressed

Software Problems Addressed

- Modular

Software Problems Addressed

- Modular
- Scaleable
  - Separate database from application
  - Separate application logic from parsing

Software Problems Addressed

- Modular
- Scaleable
  - Separate database from application
  - Separate application logic from parsing
- Maintainable
  - MVC Architecture
  - HTML, CSS, JS are separate from the logic

Software Problems Addressed

- Separate customer data from web server

Software Problems Addressed

- Separate customer data from web server
- Use REST as middleware

Software Problems Addressed

- Separate customer data from web server
- Use REST as middleware
  - Well defined vocabulary

Software Problems Addressed

- Separate customer data from web server
- Use REST as middleware
  - Well defined vocabulary
  - Easy to implement

Software Problems Addressed

- Separate customer data from web server
- Use REST as middleware
  - Well defined vocabulary
  - Easy to implement
  - Already-existing mechanisms for routing, authentication

Software Problems Addressed

- Security

Software Problems Addressed

- Security
  - Django handles sanitization

Software Problems Addressed

- Security
  - Django handles sanitization

Software Problems Addressed

- Security
  - Django handles sanitization
- Expandable
  - Additional modules can be created as-needed

Software Problems Addressed

- Security
  - Django handles sanitization
- Expandable
  - Additional modules can be created as-needed
  - Separate model, view, controller

Software Problems Addressed

- Security
  - Django handles sanitization
- Expandable
  - Additional modules can be created as-needed
  - Separate model, view, controller
  - Different servers for processing and responding to users

Software Problems Addressed

- Security
  - Django handles sanitization
- Expandable
  - Additional modules can be created as-needed
  - Separate model, view, controller
  - Different servers for processing and responding to users
    - Allows the server to process multiple applications

Hardware Problems Addressed

- Security
- Scalability

Hardware Problems Addressed

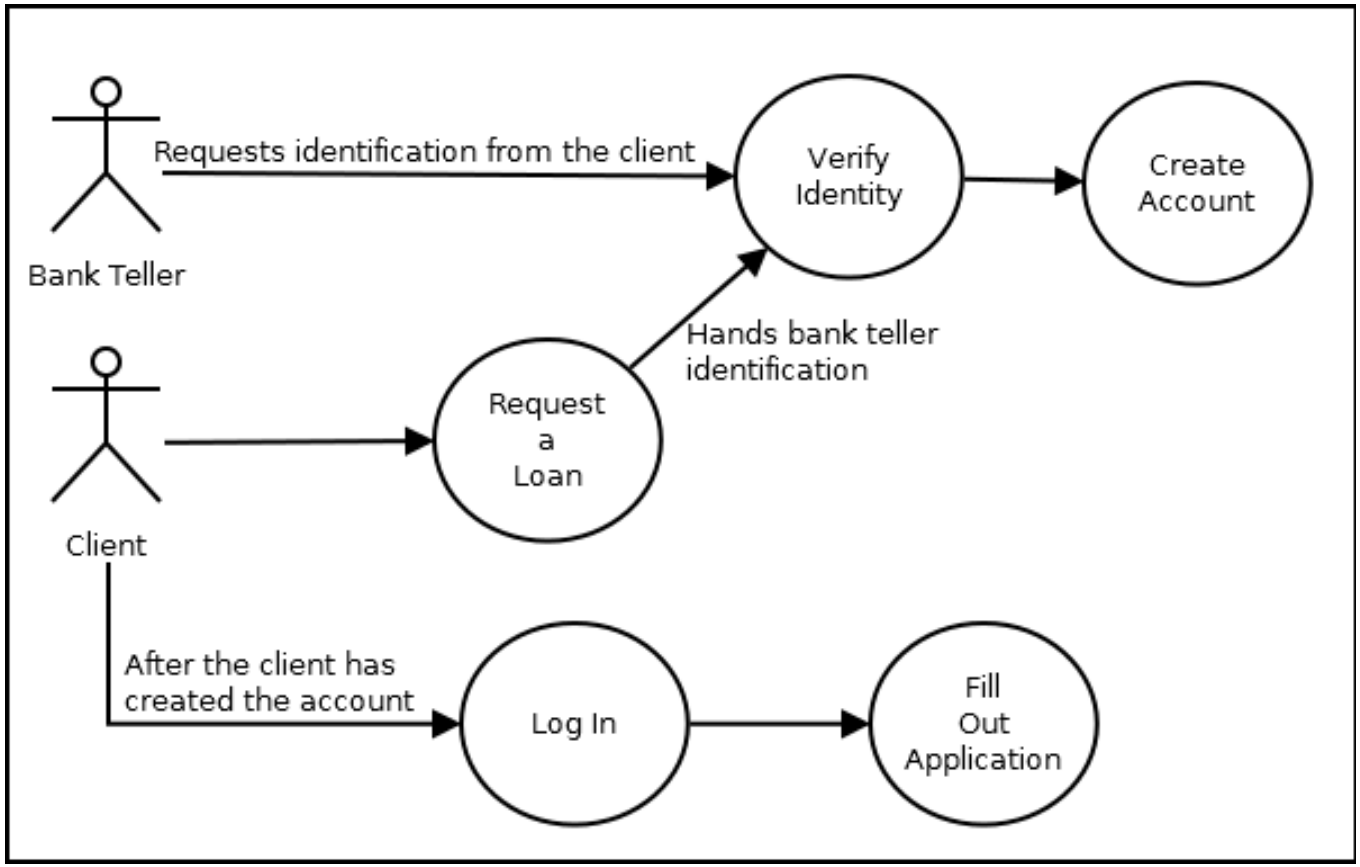
- Security
  - Routers
  - iptables
  - Seperate Internet connections
  - Seperation of execution
- Scalability

Hardware Problems Addressed

- Security
  - Routers
  - iptables
  - Seperate Internet connections
  - Seperation of execution
- Scalability
  - Routers
  - Seperation of execution

Sample Implementation

Example Use Case: Loan Request as a New Borrower



Questions?