MICHIGAN STATE UNIVERSITY

Project Plan

Parking Allocation and Expense Reconciliation

The Capstone Experience

Team Quicken Loans

Tim Hasselbeck
Chelsea Bridson
Matt Soulliere
Elizabeth Florian

Department of Computer Science and Engineering
Michigan State University

Spring 2015

From Students…
…to Professionals

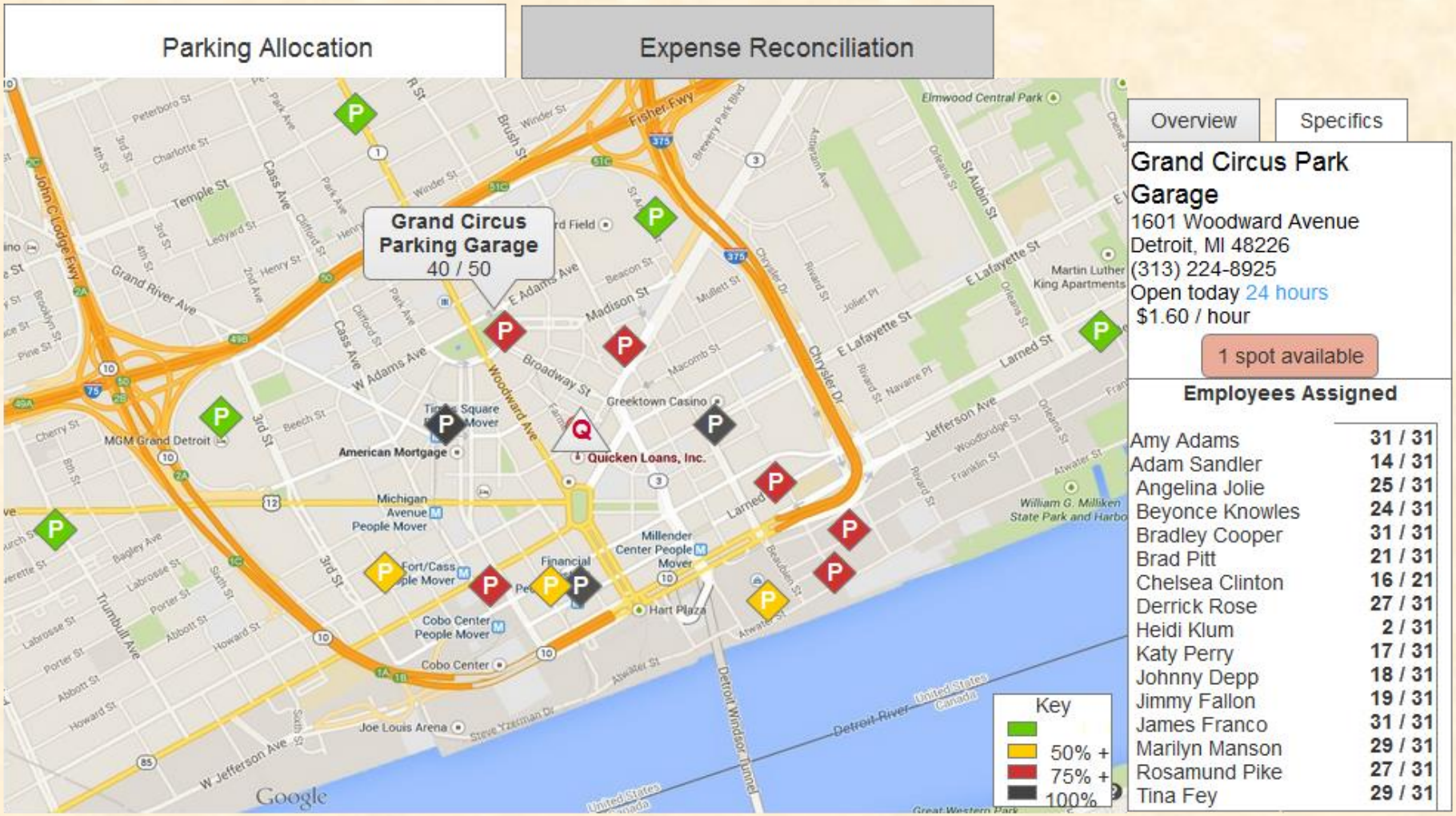
Functional Specifications

- The system will display a visual representation of the parking garages information on a map
- The system will ingest data from multiple garages and display data in a graphical form
- The system will compare and display discrepancies in parking garage data
- The system will detect employees who haven't been using their spots

Design Specifications

- Easily view used parking spaces VS. the capacity of parking garages and indicate when a garage is full
- Easily view how often a specific team member uses their allocated spot
- Indicate that garages are accurately billing for the exact number of team members allocated
- Display discrepancies between billing versus enrollment

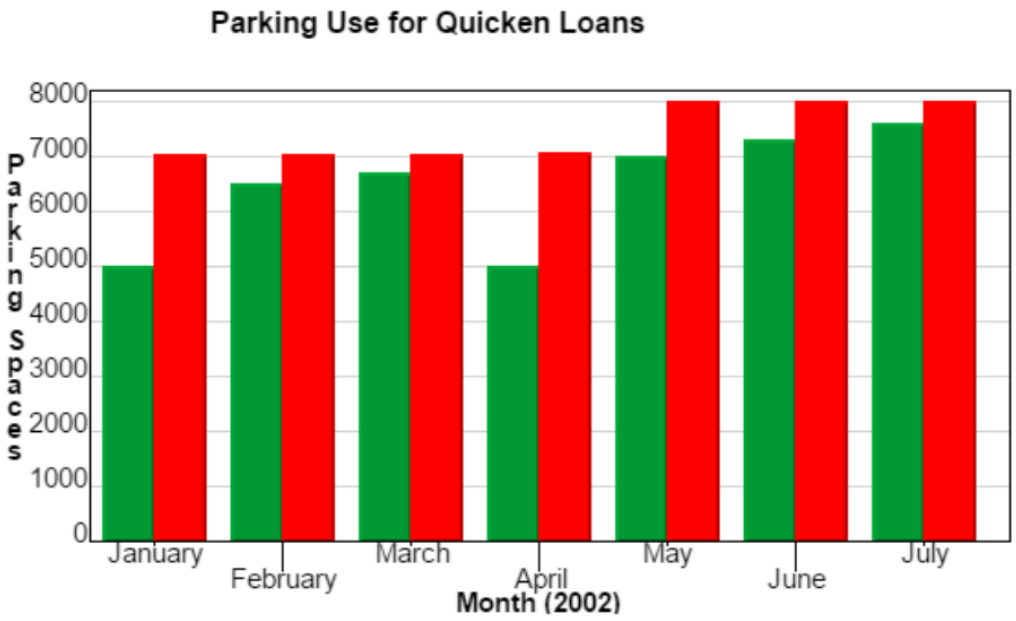
Screen Mockup: Parking Allocation



Screen Mockup: Expense Reconciliation

Parking Allocation | Expense Reconciliation — tabs

Quicken Loans Expense Reconciliation



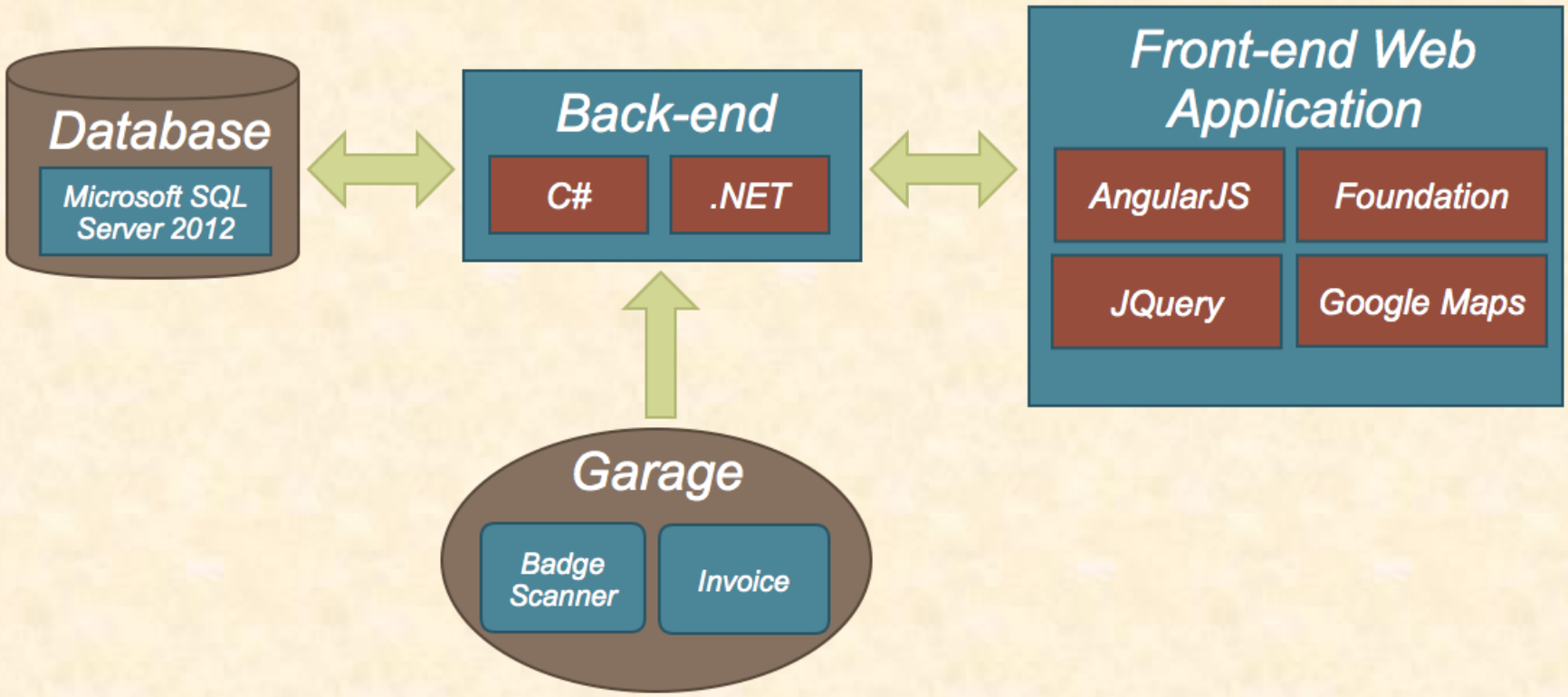
Overview | Specifics

Grand Circus Park Garage	49/50
Park Rite Trolley Plaza	31/60
Greektown Casino Parking	19/20
Handy Parking Inc	40/40
River East Parking Garage	50/50
Parking Lot A	11/40
Port Atwater Parking Garage	30/30
Parking Lot B	15/25

Technical Specifications

- Database: The database will store badge scan data, garage invoice data, Quicken Loans enrollment data, & human resources database
- .NET Framework
- ASP.NET Model View Controller – Provides a layer to interact with the database and the frontend
- Google Maps API: Provides basic map services for the Parking Allocation module

System Architecture



System Components

- Hardware Platforms
  - Badge Scan Readers
  - The Server
  - Client PC
- Software Platforms / Technologies
  - Microsoft SQL Server 2012
  - .NET Framework and ASP.NET Web API
  - Google Maps API
  - AngularJS

Testing

- Microsoft Unit Test Framework
- Ensure invoice data is being read from files correctly
- Ensure badge scan data is being read from files correctly
- Create a simulation of the Quicken Loans HR database
- Ensure calculations of discrepancies between badge scan and enrollment data are correct

Risks

- Ingesting invoice data – multiple formats for 19 garages
  - Pulling data from garages may be complicated
  - **Mitigation**: allow manual additions of invoice data
- Ingesting parking garage swipe-in data for team members
  - Different formatting for different garages
  - **Mitigation:** develop tools to identify column names
- Linking with the Human Resources Database
  - Pull employee information such as IDs and badge numbers
  - **Mitigation**: set up a test database with SQL Server 2012
- Selecting too many features, feature creep
  - Many of the features are beyond the scope of our project
  - **Mitigation**: Determine the relative importance and difficulty of each feature