Presentation overview

- Brief introduction to your client and the product to be developed
- The project results – including things like:
  - What were the client requirements? (You should not go through each requirement in detail, but stay at a high level)
  - How many (and which) requirements are fulfilled? Do you fulfil the use cases you have promised?
  - How did you prioritize in case you didn't have time to implement everything?
  - What are the high-level design (architectural) decisions? How is the program structured? How do you use the available technology?
  - How did you ensure the important qualities (e.g. real-time behaviour, security)?
- Result demonstration. This means showing your application, describing its features in a pedagogical and convincing way.
- The project work – you should present things like:
  - Activities and worked hours summarized.
  - How much of the initial plan held – how did you manage changes along the way?
  - How did you divide and synchronize the work?
  - Did the organization and routines change during the project?
- Experiences: What did you learn from the project – you should present things like:
  - What is your experience from working in a group? What were the problems (please do not be too detailed or personal)?
  - What turned out as you expected? What will you improve the next time?

Decision Tree Engine App

Software Engineering Project Teamwork

ROBOT
APPLICATION
CENTER

Group 3

Project Overview

Robot Application Center developed a decision tree engine to help partners deciding if a particular investment is profitable or not. Currently they use a desktop application to generate surveys, and through this application they deploy a final-user application to answer these surveys.

Client overview

Client details:

- Andreas Runfors - Primary client (Application owner)
- Erik Hellström - Secondary client/proxy (Robotic Application Center)

Project Team/Client Interaction:

- Meetings - questions prepared beforehand
- Email

Project Goals

Develop two applications:

- An admin web application to create general decision trees.
  - Allow admin to create/modify/delete decision trees.

- An end-user mobile app to let final users answer these trees.
  - Allow end-users to generate a summary to evaluate if their investments will be profitable.

Project Results

Client Requirements:

- Offline
- Platform independant
- Language options
- Results should be emailed to the user as a pdf

Project Results

Requirements Achieved

- Language options
- Platform independant
- Result Emailed
- Works Offline

Prioritization

- Core functionality

High-level design

- Database

  Where trees/surveys are stored
  Parse database (on the cloud)

- End-user app

  Fetches trees/surveys from the database
  Cordova (hybrid between web app and native mobile app)

- Admin (survey builder) app

  Creates and modifies trees/surveys and stores them to the database
  Web application that runs on a local computer

Program structure

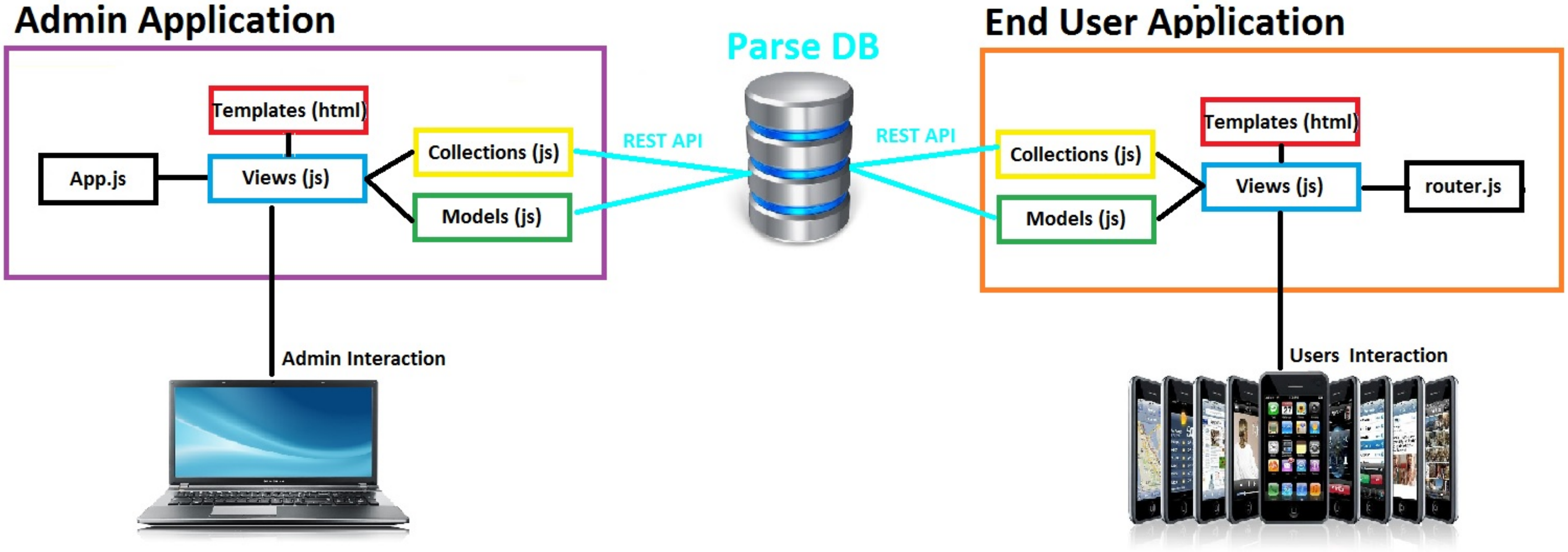
- Survey/tree structure:
- Survey/tree
- Question/node
- Answer/edge

- Economic survey/tree structure
- Calculation/formula
- Inputnode (e.g. capital, cash flow)

Program structure

- Backbone.js
- Models
  Tree, question, answer etc.
- Collections
  Collections of trees, questions, answers etc.
- View
  User interface. Handles user inputs, updates models.
  Combination of view and controller in MVC.

Program structure



Technologies

- HTML
- CSS
- JavaScript
- Backbone.js
- jQuery
- Parse database
- Cordova
- Cross-platform tool for mobile applications (hybrid)

Important qualities

- Usability
  - Testing
  - Client communication
- Confidentiality
  - No end-user data is saved on the public database.
- Non-proprietary tools
- Platform independence
  - Cordova, web technologies (HTML, CSS, JavaScript)

DEMO

Project Work

Roles

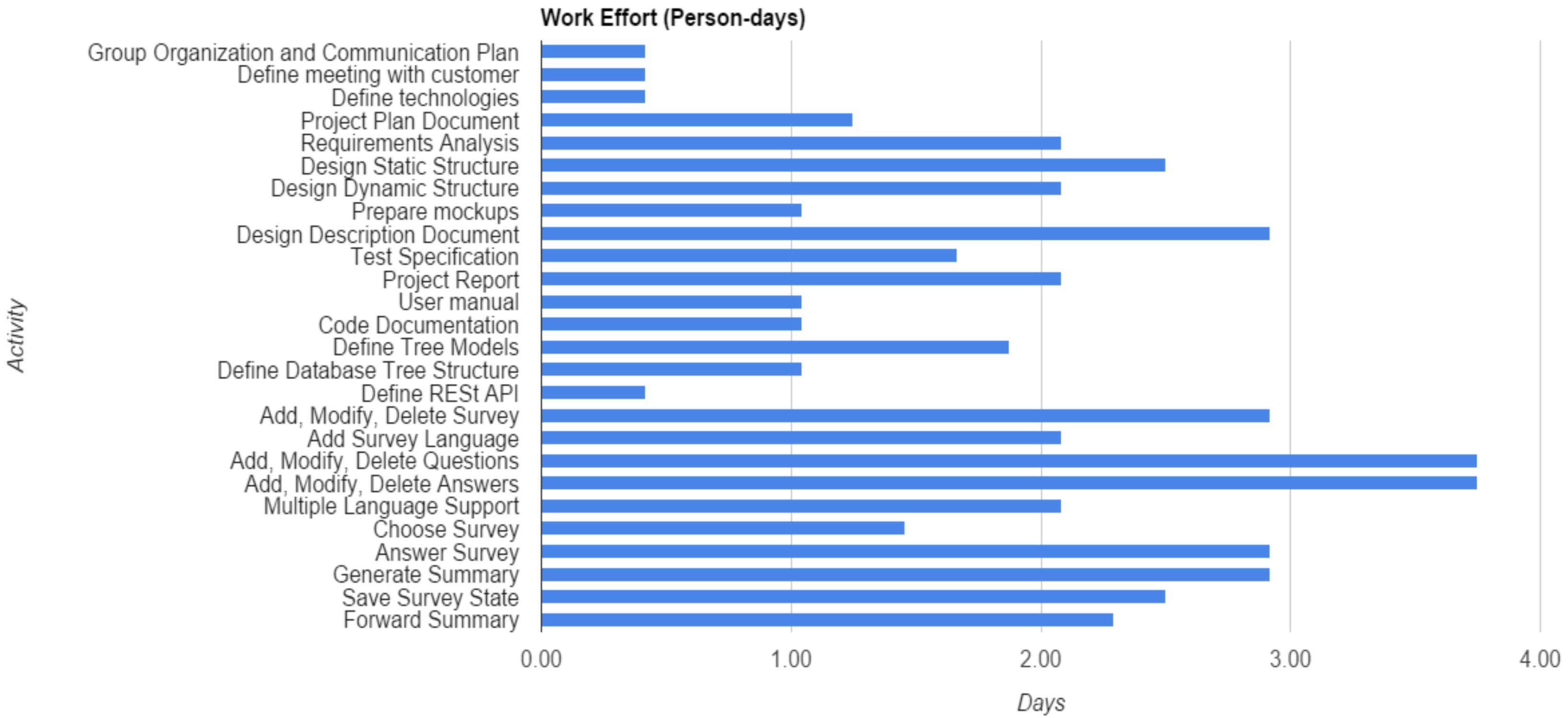
Role	Team member
Project Manager	Valerio Lucantonio
Client Liaison	Anna Enbom
Github Supervisor	Tamara Dancheva
Document Supervisor	Nathan Chape
Testing/Usability Supervisor	Niklas Sjöqvist
Testing/Usability Supervisor	Eric Johansson

Project Work

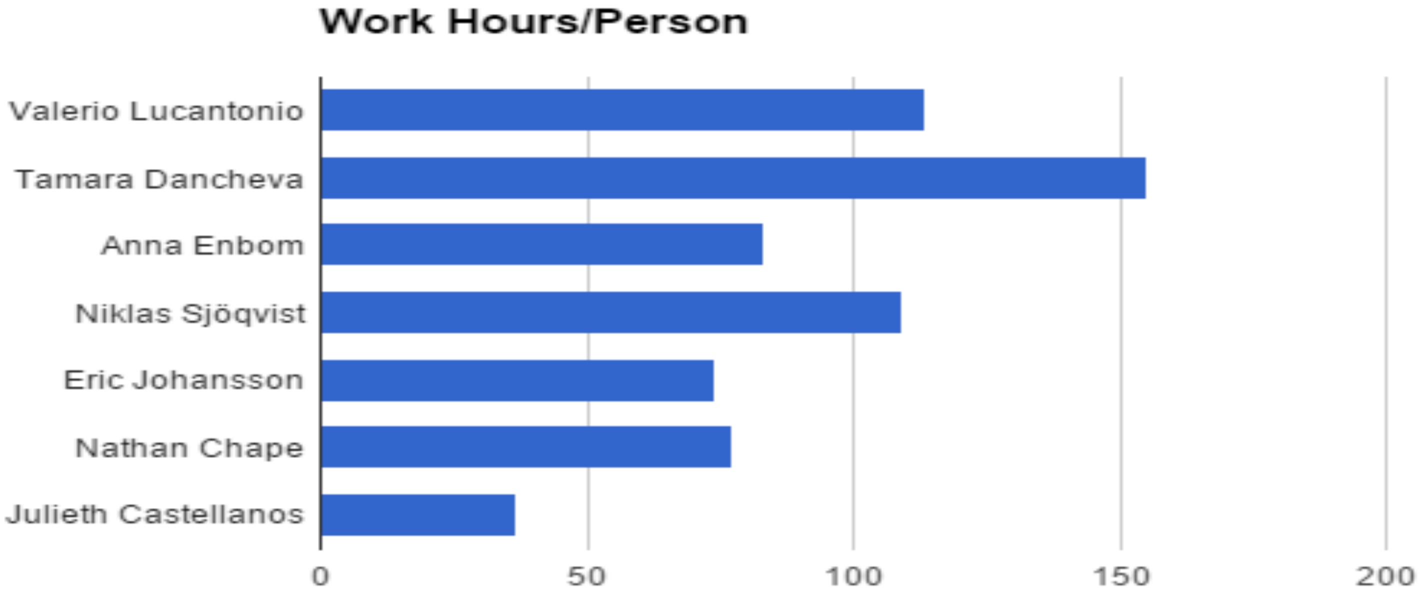
Organization and Routines

- At *least* 1 weekly team meeting
  - discussing what has been completed
  - planning each weeks work/sprint
  - usually 2 or more
- Constant communication

Project Work



[redacted]


Work Hours/Person
Valerio Lucantonio
Tamara Dancheva
Anna Enbom
Niklas Sjöqvist
Eric Johansson
Nathan Chape
Julieth Castellanos
0
50
100
150
200

Project Work

Division of labor/Subgroups

- 2 main phases
  - requirements/planning
  - implementation
- 3 subgroups working in parallel

Project Work

Role	Team members
Admin App	Anna Enbom, Nathan Chape
End User App	Tamara Dancheva, Niklas Sjöqvist, Eric Johansson
Database/App interface	Valeritonio Lucan

Project Work

Changes

- Loss of team member

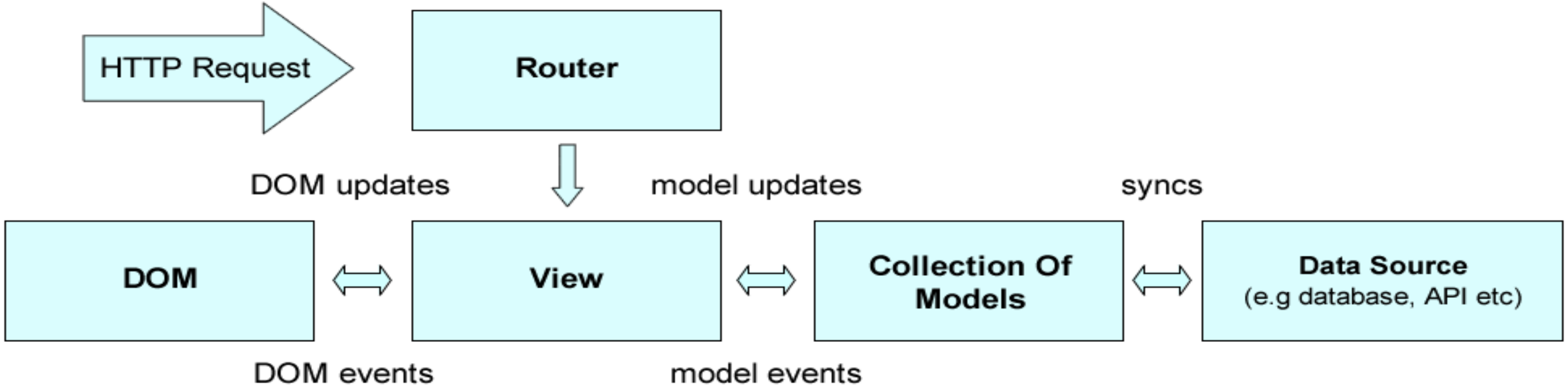
Experiences

- Document, document, document
- Working with a diverse group
- Working with a large project

Questions

Main design decisions

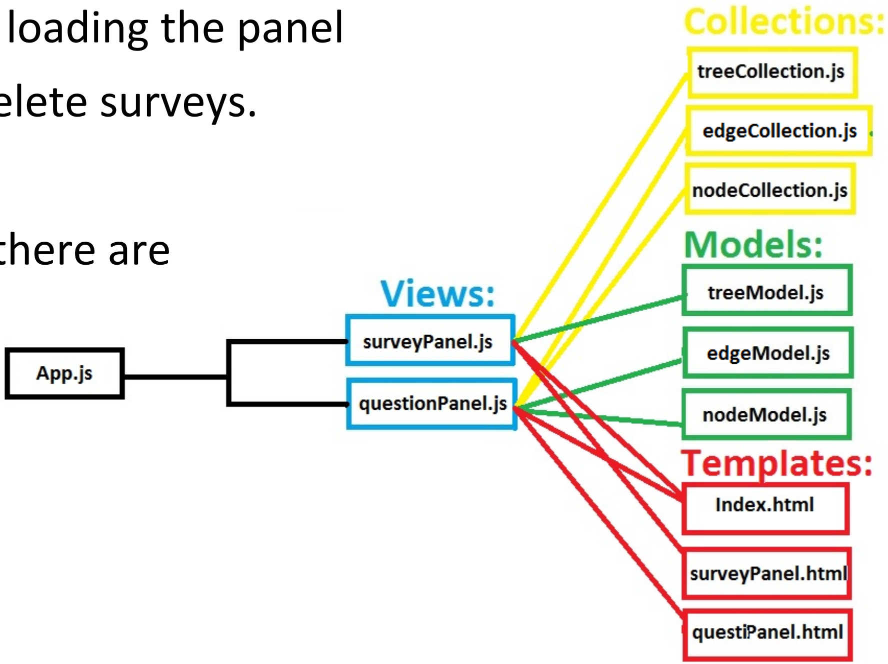
- Using Backbone.js for both the apps (MVC architecture)
- Shared Online Database (PARSE)
- Shared Models (API)



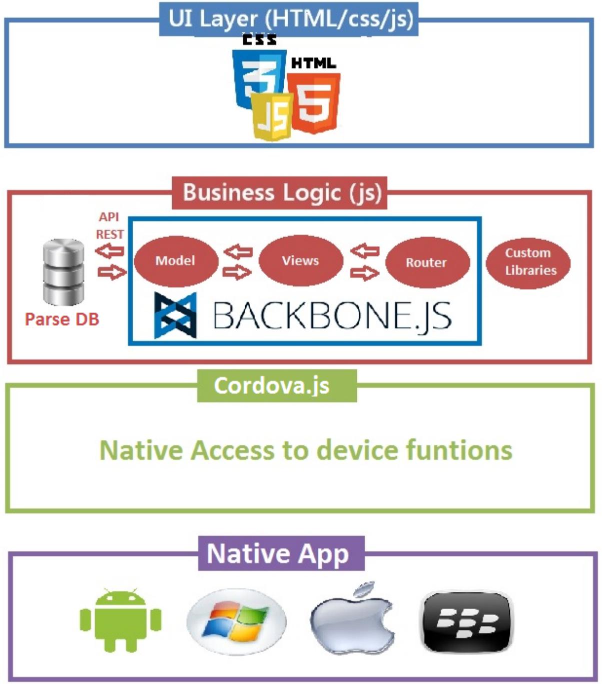
Administrator application

surveyPanel.js: the class responsible for loading the panel where the admin can add, modify and delete surveys.

questionPanel.js: in the question panel there are forms to create, modify and delete questions and related answers.



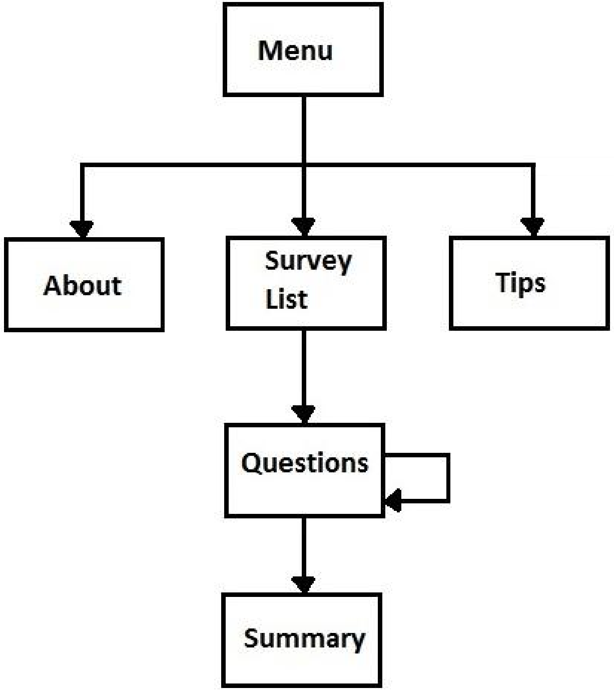
End-User Application



End-user Application

Main functionalities:

- Be able to work offline
- Generate a summary after the survey is completed
- Save state of answered questions
- Contact administrator
- Multiple languages support


Menu
About
Survey List
Tips
Questions
Summary