Task Roulette

High Level Design

COP 4331, Spring, 2014

Team Name: MADNESS (aka. Team 14)

Team Members:

- Cody McMahon
- Jessica Carter
- Matt McGivney
- Steven Lo
- Gunnar Skotnicki

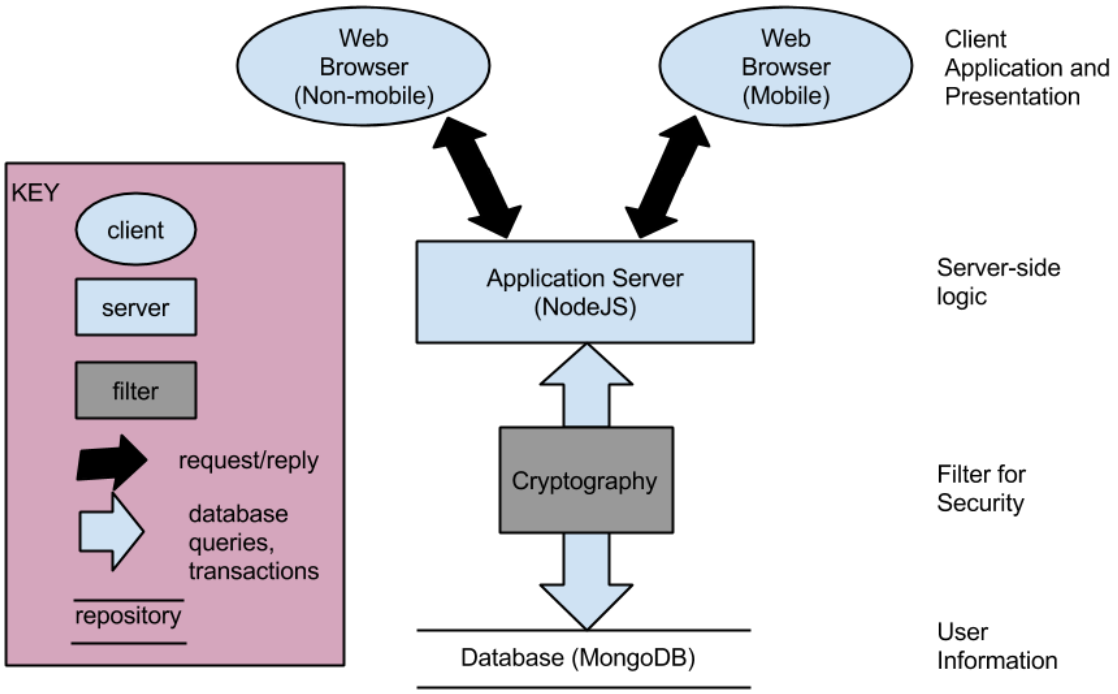
Modification history:

Version	Date	Who	Comment
v0.0	05/13/13	S. Applegate	Template
v1.0	02/18/14	C. McMahon	Begin Format Changes
v1.1	02/27/14	J. Carter	Design Issues
v1.2	2/27/14	S. Lo	Design Issues
v1.3	2/27/14	C. McMahon	High Level Architecture diagram and Design Issues
v1.4	2/27/14	M. McGivney	Design Issues

Contents of this Document

Section 1: High-level Architecture

CLIENT SERVER:



The Application Server (Node.JS) will handle the requests from the user. Following the request it will encrypt user credentials as well as any other necessary information.The information is then processed into the database and stored for later retrieval or Node.JS will handle the requests to get information from the database and will serve the requests to the webpage.

Section 2: Design Issues

Reusability**:**

The way we are creating this application with Node JS, Angular JS, Bootstrap, and MongoDB will make it easy for us to create other applications with this code. TR could easily be transformed into other projects such as a web-based bookkeeping application or a web-based note taking application.

Maintainability:

Separating our application with a MVC (Model, View, and Controller), we can easily maintain the design requirements with the functional requirements. We are using current technologies that are supported by companies like Google.

Testability:

The independence of our models allows for easy testing. Each module can be extensively tested on its own. Afterwards, components can be combined and tested together until the entire software system is composed and tested as a whole. This method creates easy and thorough testing of the system.

Performance**:**

Performance is not an issue since we are using Node JS for a web-based application. This framework combined with Bootstrap and Angular JS provides a lightweight system where performance requirements are a minimum.

Portability:

Task Roulette will be able to run on mobile and non-mobile devices since we are using Bootstrap. It configures its display format based on the user's screen resolution, allowing Task Roulette to display correctly on any screen size.

Safety:

Task Roulette is not responsible for the tasks each user inputs. If a user inputs an illegal task, it is entirely the user's responsibility.

Connectivity**:**

The user will need to be connected to the internet in order to access our application.

Section 3: References

- https://sites.google.com/site/cop4331group2/high-level-design
- https://sites.google.com/site/cop4331team4/high-level-design
- http://ucf.karlbanks.com/cop4331/deliverables/del2/#del2_1