Requirements Analysis Document

Massive Dynamic:

Liron Mizrahi - 708810

Marko Vidalis - 831987

Jason Chalom - 711985

2016

Executive Summary

Massive Dynamic (MD) proposes a system which will help students trade, borrow and sell second hand books – mainly textbooks. The system's main objective is to allow users to easily search for, locate and purchase the books through it. It will have gamification aspects like Uber where users can rate other users. Additionally, it will include a reporting function in order to allow reporting of abusive users. There will be an administrative profile/s in the system, to aid for the maintenance of the system. The technology we have chosen to implement this on is a web based technology, namely the MEAN stack. The system will not cover using online payment methods and will only facilitate the locating and transferral of physical goods.

1. Introduction

1.1 Purpose of the system

MD is focused on building a system that will help facilitate the borrowing and/or selling of textbooks and other books between students. The system will not allow the transfer of copyrighted materials but will rather facilitate the lending and selling of second-hand textbooks. i.e. trading.

If a user has found a textbook which they would like to purchase, the seller is then contacted through contact details provided through the system and the purchaser can then buy the desired book.

The system will have a gamification function which will allow users to rate each other, similar to how the Uber app works. This rating will represent how users behave in their communications, transactions and reliability in the process of trading through the system. Any transactions made will be logged, which will allow the system to provide a history to the user.

Users can rate each other after a transaction is completed and also report users who abuse the system. These 'bad' users can then be suspended from the system, which is an administrative feature.

1.2 Scope of the system

The system will not include a mobile app. It will also not include any complex login profiles or file transferring. However there will be secure login with password salting and hashing. Besides regular user logins, there will also be admin logins with extra privileges.

It will consist of a web server and an html browser front-end. It will be a cash based trading platform and any transactions done will be in person. There will also be a post-transaction rating system in which other users are rated according to how well the transaction went. There will also be a reporting feature where a user may report another user. The system will also have a bug reporting system where users can report any bugs found on the site.

1.3 Objectives and success criteria of the project

- Secure login credentials using password salting
- Ability for admin to manage users
- Ability for users to communicate and trade books
- Ability to rate users
- Ability to report users

Measures of Success:

Efficiency:

The system must have efficient work flows for users and admins.

Responsiveness:

The system must be responsive and perform well whilst delivering content to multiple users.

Quality:

Users and stakeholders must be happy with what is delivered every sprint and the final deliverable.

2. Current system

The current system of obtaining any books or textbooks for students is by going to either a library or to a book store, for both first and second-hand books. This means that there are advantages to the sellers of books as there are many more buyers in comparison, due to the fact that there are limited centralised locations at which a specific book may be found. However it is not always easy to find the exact book a customer is looking for. While this current system has worked in the past, there are very few receipts and logs of transactions made, which makes tracing any purchases very difficult. Finding books that a specific student is looking for is also very difficult and time consuming and this is what MD serves to solve through this system.

3. Proposed system

3.1 Overview

The system will consist of a server which will serve webpages to users and also process user requests. EJS will be used as the templating engine which will dynamically render webpages to the user. MongoDb will be used for the database backend. User information, including salted passwords, transaction logs and any other functional system information will be stored here.

3.2 Functional Requirements

3.2.1 As a user, I can create an account on the system.

3.2.2 As a user, I can login to the system.

3.2.3 As a user, I can change my password.

3.2.4 As a user, I can explore books other users have posted to the system.

3.2.5 As a user, I can post books Im willing to trade or loan on the system.

3.2.6 As a user, I can post books Im willing to sell on the system.

3.2.7 As a user, I can rate my experience during a specific transaction.

3.2.8 As a user, I can report a user who has abused the system.

3.2.9 As an Admin, I can reset user passwords.

3.2.10 As an Admin, I can see and review which users have been reported for bad behaviour.

3.2.11 As an Admin, I can reverse transactions when something unexpected happens in the system.

3.2.12 As a user, I can contact admin on the system.

3.2.13 As an Admin, I can see messages sent by users and respond to them.

3.3 Nonfunctional requirements

3.3.1 Usability

The system must have a pleasing interface which is simplistic in design and intuitive to use.

3.3.2 Reliability

The system must support multiple users simultaneously and not be prone to hanging or crashing during operation.

3.3.3 Supportability

The web based approach allows the system to be opened by any HTML5 supporting web browser. This enables the system to run on many different devices and device eco-systems.

3.3.4 Implementation

The server backend of the system will be built using the MEAN stack, which includes MongoDb for the database, nodejs for the server backend, and ExpressJS for the frontend. We want to also use PassportJS to help us with user logins. Our tests will be done with Mocha and Chai libraries. WinstonJS will be used to generate server logs to help us maintain server operation.

4. Glossary

- MEAN stack: A software bundle used in making dynamic web systems and information systems. It includes NodeJS, ExpressJS, AngularJS and MongoDb.
- PassportJS: A library for NodeJS which is used for OAuth and user accounts on a node system.
- ExpressJS: A library for NodeJS which provides dynamically generated frontends and RESTful APIs.
- EJS: A templating library and language used with ExpressJS for web page generation.
- Templating Engine: A tool used to separate program-logic and presentation into two independent parts.
- WinstonJS: A logging library used to make system logs for NodeJS.
- MongoDb: A NoSQL or object oriented database which uses JSON objects and RESTful Api calls.
- NodeJS: A runtime environment which is used to build server side applications in Javascript.
- Mocha and Chai JS: These are unit and business process testing frameworks (libraries) for NodeJS.
- Node system: A system built on NodeJS.

5. Team

- Our Team name is: **Massive Dynamic**
- The Scrum Master: Liron
- The Project Owner: Jason
- Software developer/designer: Marko

Appedix A: Statement of Effort

	Jason Chalom 711985	Marko Vidalis 831987	Liron Mizrahi 708810
Finishing	0%	100%	0%
Introduction	33%	33%	33%
Current System	0%	20%	80%
Proposed System	75%	0%	25%
Glossary	90%	10%	0%