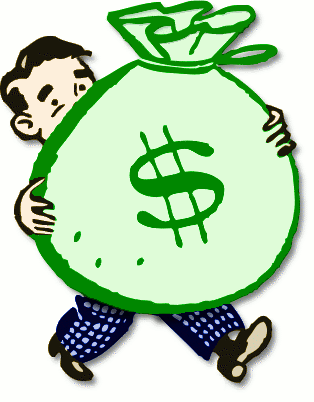
**“Money On The Go” SMART PHONE APPLICATIONS**

***PROJECT MANAGEMENT~ PartII-B***

*IS663 - Team Project, NJIT*

****By **Money On The Go’s Team**:

Ayodeji Ojo

Awad Albalawi

(*Team Leader*)Blandine Meillon

Isaac Osobu

Rotimi Oduntan

Teko Folligan

**Table of Contents**

1 Introduction 3

2 Project Organization 3

3 Risk Analysis 5

4 Hardware and software Resource requirements 6

5 Work Breakdown & Project Schedule 7

5.1 Project Backlog Work Breakdown 7

5.2 Project Schedule 8

6 Monitoring and Reporting Mechanisms 10

7 References 12

1. Introduction

With the volatility of increasing market instability and unpredictable situations it is becoming essential to find a viable software or app that can alert and suggest to individual investors on the happenings in the stock market for decision support process. “Money on the Go” abbreviated as MOG is a smart phone web application use to track and/or manage investment the user owns

across every device of choice. It works by promptly pulling all users’ investments intoa single

dashboard in order to see the bigger picture, eliminates unnecessary fees, increase profitability,

and helps fix underperforming investments. Various investments might be bank accounts, 401k

plans, IRAs, stock holdings, and the like; while giving tailored recommendations about use portfolio’s performance and changes that user could make.

The aim of this project is to design an App that will act like an investment advisor. It will enable users get high return on investments by utilizing a proprietary web-based technology platform. MOG provides non-discretionary investment recommendations and personal investment related data to individuals (the “Clients”) based on such Client’s trading patterns and existing individual portfolio investments. MOG’s advisory services are made available to Clients through its website at www.MoneyOntheGo.com, mobile apps and through co-branded pages and widgets on its media partners’ websites (collectively the “Website”). In all cases, including those where the services are made available through a co-branded partner website, MOG’s web-based platform is powering the investment advisory services and hosting client data.

2. Project Organization

The project organization focuses on the software process model which is a set of correlated actions that leads to the making of a software product. In short term, it is a streamlined illustration of a software process. Scrum Agile method will be best suited process model for this project.

Scrum is an iterative and incremental agile software development method for managing software projects, product and application development. According to Schwaber (2011), “It has been used to manage complex product development since the early 1990s. Not a process or technique for building products; rather, it’s a framework within which you can deploy various processes and techniques.” It is a process within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value. It consists of time boxed iterations and each iteration produces some real software that has value to the users and clients. It also focuses on lowering risks and delivering highest returns on investment in shortest time by rapidly creating and repeatedly inspecting real working software.

The iteration process has iteration planning, development work and iteration review. The iteration planning involves selecting next features, define and estimate tasks, negotiate scope of the delivered product. Iteration execution implements the items in the plan and the results of the iteration can be delivered in a demo. Scrum elements consists of three different roles namely product manager (owner) – Prof. Kirova, scrum master – Blandine Meillon and team member (Developer) – The rest of the team. The product owner in this case is the project's key stakeholder and is the voice of the customer. The product owner writes customer-centric items, prioritizes them and adds them to the product backlog. The team in scrum is self-organizing and consists of people with cross-functional skills who do the actual work.

They implements items on the PBL/sprint backlog, delivering value to the customer often. The scrum master is responsible for removing impediments to the ability of the team to deliver the sprint goal/deliverables. The role involves that the scrum master prevents interference during iteration. In addition, the scrum master is both a teacher and a referee. Scrum enables the creation of self-organizing teams by encouraging co-location of all team members, and verbal communication between all team members and disciplines in the project.

A key principle of Scrum is its recognition that during a project, the customers can change their minds about what they want and need (often called requirements churn), and that unpredicted challenges cannot be easily addressed in a traditional predictive or planned manner. Scrum iteration is called a sprint. The sprint is a "time boxed" effort, i.e. it is restricted to a specific duration. The duration is fixed in advance for each sprint and is normally between one week and one month. This ensures an appropriate amount of time is spent planning without allowing waste in the planning process. The daily scrum meeting is conducted to keep track of daily project progress. During sprint review meeting, the scum master and the product owner access the product and provide their feedbacks for either new changes to be incorporated to the product or the product release. The three main roles from Scrum Elements concerning “MOG” as identified above are as this:

**2.1 Project Team & Roles**

* *Product Manager (Owner):* Prof. Kirova

Responsible for maximizing the value of the product and the work of the Development Team. Also sole manager of the Product backlog

* *Scrum Master:* Blandine Meillon

Responsible for ensuring Scrum is understood and enacted, that Scrum team adheres to Scrum theory, practices, and rules. It is a servant-leader for the Scrum Team.

* *Team Member (Developers):* Consist of professional who do the work of delivering a potentially releasable Increment of “Done” product at the end of each Sprint.

|  |  |
| --- | --- |
| Ayodeji Ojo | *Program Management* |
| Awal Albalawi | *Offer Management* |
| Rotimi Oduntan | *Product Management/Testing* |
| Teko Folligan | *Systems Engineering* |
| Issac Osobu | *Architecture* |

3. Risk Analysis

The below table has the risk management which includes:

* Risk identification
* Risk analysis
* Risk Planning
* Risk monitoring

**Table 3.1 Risk Management**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Risk Type* | *Possible Risks* | *Effects* | *Probability* | *Strategy* | *Potential indicators* |
| Technology (1) | Security breach and data loss can emerge when customers’ accounts are hacked by malwares and viruses and connect to MOG network  Possible security breach when syncing with customer financial institutes | Catastrophic | High | Encrypted the system with 256-bit SSL security  Use SSO (Single Sign On) which is a blind verification process what will authenticate the user inputs and link it to the appropriate financial institute.  Use a 60 seconds timer that will verify all authorized processes constantly running on the MOG network | Three unsuccessful attempts to login into customers’ account  Unexpected shut down of the system  IDM (Intrusion Detection Monitor) detect malicious software or unauthorized access to the network |
| People (2) | Key programmers and decision makers are sick and unavailable during critical time  Skilled personals are hard to find for a short period of time | Serious | Moderate | Reorganize team and decision makers so there is more overlap of work and people therefore understand each other’s job function | Poor communication and relationship among team members  High staff turnover |
| Estimation (2) | The time required to release the software application is underestimated  The size of the software is underestimated | Tolerable | Moderate | Alert customers and stake holders of potential difficulties and possibility of delays  Provide assistance to all developers falling behind  Investigate bringing in qualified consultants for short period of time | Failure to deliver PBL during iteration  Failure to clear reported defects  Failure to meet agreed dead line |

4. Hardware and Software Resource Requirements

In this section, the document will lay out hardware and software configuration, as well as the resources/developer concerning this application.

**4.1. Hardware Configuration:**

This aspect will focus on the enterprises architectures of the development environment, testing and storage machines. MOG architectures will be design using Microsoft and Java technologies and Amazon Multidimensional Data Storage (Data Repository). Applications developed in these disparate technologies will communicate seamlessly with each other. Data transfer protocol technologies like HTTP, WEB Service and TCP should be carefully utilized to enable efficient flow of data throughout the entire enterprise system. Lastly, the hardware will involves storage capabilities like Database in Terabyte sizes for analysis purposes.

**4.2. Software**

MOG software will focus on tools that are user friendly. MOG software platform generates non-discretionary recommendations on the basis of an analysis of investment data; the Client’s trading patterns and the Client’s existing portfolio investments. The software will focus on of Graphical User Interface (GUI) and small physical characteristic nature of mobile devices using Java script and ASP.Net MVC. The application will be web-based and will be built to run on any modern web browser on any platform. A mobile application will normally be structured as a multi-layered application consisting of user experience, business, and data layers. The Java 2 Platform, Enterprise Edition (J2EE) is a specification delivered by Sun Microsystems using the Java Community Process (JCP) will be used to develop the app. The J2EE specification defines a standard platform for building multitier enterprise applications. It builds upon capabilities provided in the Java 2 Platform, Standard Edition (J2SE), such as the core Java libraries, the promise of code portability, and Java Database Connectivity (JDBC).

**4.3. Resources – Developers**

* Java Developers
* ASP.Net Developers
* Mathlab & SAS Developers (QA)
* Quantitative Analyst
* Testers

5. Work Breakdown

In this section, the product backlog will be the focus; It evolves as the product and the environment in which will be used evolves. The product backlog consists of all features, functions, requirements, enhancements, and changes that will made to the product future releases.

**5.1. Product Back Log Work Breakdown**

**Table 5.1 – Work breakdown**

|  |  |  |  |
| --- | --- | --- | --- |
| No. | User Story | Priority (1-4) | Storylines |
| 1 | As a user, I need to log in/register while keeping my information safe. | 1 | 7 |
| 2 | As an app user, I can create an account, add and delete many accounts to my profile. | 1 | 2 |
| 3 | As a user, I want to Sync all accounts in single dashboard. | 1 | 5 |
| 4 | As a user, I want to see all hidden fees | 1 | 3 |
| 5 | As an app user, I can update my accounts. | 2 | 2 |
| 6 | As a user, I want to have real time update (new charges on accounts, new fees & Taxes). | 2 | 5 |
| 7 | As a user, I want to receive a recommendation | 2 | 7 |
| 8 | As an app user, I can view my investments in different categories (chart & tools). | 2 | 4 |
| 9 | As a user, I want to read more about the app’s features. | 2 | 3 |
| 10 | As a user, I want to have a weekly summary. | 3 | 2 |
| 11 | As a user, I want to be able to select automatic update. | 3 | 2 |
| 12 | As a user, I can customize my dashboard. | 3 | 6 |
| 13 | As an app user, I want to modify my profile page to fit my style. | 4 | 3 |
| 14 | As an app user, I can rate and send reviews/feedback about the app. | 4 | 4 |

**5.2. Project Schedule**

In this section, the document will lay out a SWOT analysis, emphasize on the selected process model, user stories through product backlog, project plan, and in the next sections, elaborate on risk management involve with this plan and integration with other brokerage companies.

**Table 5.2. Work Schedule Breakdown by Hierarchy**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Level 1 | Level 2 to 3 | Time Est. (Days) |
| Money On The Go App | **Phase I** | 1.1 Evaluation & Analysis of Requirements  1.2 Gather Data – Functionality  1.3 Design the Module  1.4 Create Code & Design for  1.4.1 User Login and Registration  1.4.2 Add account to Profile  1.4.3 Sync all account to a single dashboard  1.4.4 See all hidden fees  1.1.5 Code Review & Unit Testing  1.1.6 Rework if Any  **SPRINT 1** | 2  2  2  -  7  2  5  3  2  2  **27** |
| **Phase II** | 2.1 Verify & Analysis User Requirements  2.2 Gather Data – Functionality  2.3 Coding & Design  2.3.1 Update accounts  2.3.2 Real time update  2.3.3 Receive recommendation  2.3.4 View investments via charts/tools  2.3.5 Read more about the app’s features  2.4 Code Review & Unit Testing 2.5 Rework if Any  **SPRINT 2** | 1  2  -  2  5  7  4  3  2  2  **28** |
| **Phase III** | 3.1 Verify & Analysis User Requirements  3.2 Gather Data – Functionality  3.3 Coding & Design  3.3.1 Weekly summary  3.3.2 Select automatic update  3.3.3 Rate/view feedback  3.5 Code Review & Unit Testing  3.6 Rework if Any  **SPRINT 3** | 1  2  -  2  2  4  2  2  **15** |
| **Phase IV** | 4.1 Verify & Analysis all User Requirements  4.2 Install Live System on Mobile 4.3 Overall Code Review & Unit Testing  4.4 User Acceptance Testing  4.5 Rework if Any  4.6 App Code Documentation  4.7 Formal Acceptance  4.8 GO LIVE & Support  **SPRINT 3** | 1  1  2  1  1  2  1  1  **10** |

6. Monitoring and Reporting Mechanisms

In this section, reporting and monitoring mechanisms will be detailed in order to show how MOG and all others elements involved managed to successfully delivered a final product.

**6.1 Monitoring will be based on the following steps:**

* *Focus on results and follow-ups*: It looks for “what is going well” and “what is not progressing” in terms of progress toward the intended results;
* *Regular communication by the project coordinator or manager*: The project coordinator or manager should be dedicated to assessing progress, looking at the big picture and analyzing problem areas. They should ensure continuous documentation of the achievements and challenges as they occur and avoid having to try to remember the events some time later;
* *Regular analysis of reports*: The project coordinator or manager should review project -related reports, including financial reports, by the implementing partners to serve as a basis for their analysis;
* *Use of participatory monitoring mechanisms to ensure commitment, ownership, follow-up, and feedback on performance*: These include outcome groups, stakeholder meetings, steering committees, and focus group interviews;
* *Ways to objectively assess progress and performance based on clear criteria and indicators stated in the logical framework matrix of the project document*: The project team should agree on a performance measurement system by developing indicators and baselines;
* Active generation of lessons learned, ensuring learning through monitoring tools, adapting strategies accordingly and avoiding repeating mistakes from the past.

**6.2 Project Monitoring Breakdown**

Project monitoring breakdown is the different ways of monitoring information that MOG used to accomplished this project. Those main ways are the following:

* Weekly review/feedback meeting with the Product Owner
* Weekly Skype meeting
* Using Wiggio.com for file sharing
* Using Google doc for file sharing/update
* Release deliverables weekly

**6.3. Project Reporting**

Project reporting is the formal presentation of monitoring information. The main reasons for reporting are the following:

* *To formally inform Product Owner*: Reporting ensures that management, particularly the project coordinator’s supervisor or project manager, is formally appraised of the progress made in project implementation and the supervisor or project manager is aware at an early stage of actual and potential problems and any remedial action taken;
* *To validate requests for further funding*: Reporting ensures that the Scrum owner and product owner is kept informed of all aspects of project implementation.
* *To serve as an audit and evaluation trail*: Reporting maintains a record of all actions taken during project implementation. It therefore constitutes a vital resource for Product Owner and testers in assessing whether a project has been implemented in accordance with the rules and regulations and as efficiently and effectively as possible;
* *To serve as a reference for future projects*: Reporting serves as a vital resource for ensuring that lessons learned (project successes, failures, best practices) through project implementation are available for consideration when formulating and implementing future projects.

**Table 6.3. Reporting requirements for projects and internally implemented activities**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Report Type | Prepared by | Responsibility | Preparation Frequency/Period | Submission | Format |
| Progress Report | Technical Lead | Lead Developer | Daily | Project Manager | Email/Stand-up meeting |
| Final Report | Technical Lead/QA | Lead Developer | Weekly | Project Manager | Email |
| Product Evaluation | Testers/Project Manager | Project Manager | Iteration process/Each deployment stages | Product Owner  Cc: Project Manager | Project SharePoint Link/Production Server |

7. References

Schaber, K. & Sutherland, J. (2011, Octover). SCRUM.ORG. “The Scrum Guide”. Retrieved on Friday 08th, 2013, from: https://export.writer.zoho.com/images/spacer.gifhttp://www.scrhttps://export.writer.zoho.com/images/spacer.gifhttps://export.writer.zoho.com/images/spacer.gifuhttps://export.writer.zoho.com/images/spacer.gifhttps://export.writer.zoho.com/images/spacer.gifm.org/Scrum-Guideshttps://export.writer.zoho.com/images/spacer.gif.

### Somerville, I. (2011). SOFTWARE ENGINEERING. “Software Process Models” & “Agile Software Development”. Pg.30, 32-34 & pg. 57-77. Retrieved on Friday 08th, 2013.