



QA Manual

Quality Assurance Manual

Document Control

Editor	Version	Date	Update
Lewis Thresh	0.1	30/01/2015	Made the QA manual template
Lewis Thresh	0.2	31/01/2015	Added section 3 – Deliverables Author: Calum Armstrong Added section 4 (excluding 4.2) – Project Plan Methodology Authors: Ali Jewers, Sam Raeburn
Lewis Thresh	0.3	31/01/2015	Added new section 4.1 with updated diagram Author: Ali Jewers
Lewis Thresh	0.4	31/01/2015	Added Section 1 - Introduction
Lewis Thresh	0.5	31/01/2015	Added Section 2 - QA job roles Authors: All group members.
Lewis Thresh	0.6	01/02/2015	Added Section 4.2 – Project Plan Author: Jake Rasom
Lewis Thresh	0.7	01/02/2015	Cited all images and figures in order and created section 5 - References
Lewis Thresh	0.8	01/02/2015	Added Section 6 - Appendix A – Document Templates Authors: Ali Jewers, Calum Armstrong, Jake Ransom, Dan Berhe, Emmanuel Olutayo Added Section 7 - Appendix B Authors: Ali Jewers, Sam Hall
Lewis Thresh, Ali Jewers	0.9	01/02/2015	Formatting in collaboration with Ali Jewers
Lewis Thresh, Ali Jewers	1.0	01/02/2015	Amended formatting
Calum Armstrong	2.0	05/02/2015	Major alteration in formatting / styling to comply with company standards Re-arranged document templates section to improve clarity
Penny Nicole	3.0	12/03/2015	Updated Section 6.9 – Weekly Financial Review Template
Alex Cash	3.1	02/06/2015	Update to remove deprecated methods and verify QA metrics
Alistair Jewers	3.2	02/06/2015	Clarified wording of Lead Software Developer QA metrics, and bug reporting process.
Sam Raeburn	3.3	02/06/2015	Updated QA metrics for testing job role.

Calum Armstrong	3.4	02/06/2015	Minor formatting around company logo, updated QA Metrics and job role for Contracts and documentation manager
Daniel Berhe	3.5	02/06/2015	Updated QA Metrics and job role
Jake Ransom	3.6	02/06/2015	Updated QA Metrics and Job role
Lewis Thresh	3.7	02/06/15	Updated QA Metrics, Job role and inserted sofia logo
Sam Hall	3.8	02/06/15	
Emmanuel Olutayo	3.9	02/06/15	Updated QA Metrics

Table of Contents

1.0 Introduction.....	6
1.1 <i>Company Profile</i>	<i>6</i>
1.2 <i>Vision.....</i>	<i>6</i>
2.0 Jobs and Responsibilities.....	6
2.1 <i>Organisation structures</i>	<i>6</i>
2.2 <i>Job Roles.....</i>	<i>7</i>
2.2.1 <i>Project Manager – Alex Cash.....</i>	<i>7</i>
2.2.2 <i>Lead Software Developer - Alistair Jewers</i>	<i>9</i>
2.2.3 <i>Lead Software Tester - Sam Raeburn.....</i>	<i>11</i>
2.2.4 <i>Contracts and Documentation Manager - Calum Armstrong.....</i>	<i>12</i>
2.2.5 <i>Specialist Software Developer - Daniel Berhe</i>	<i>14</i>
2.2.6 <i>Marketing Manager - Jake Ransom</i>	<i>15</i>
2.2.7 <i>Lead User Experience Designer - Sam Hall</i>	<i>16</i>
2.2.8 <i>Company & Application Brand Designer / Assistant User Experience Designer - Lewis Thresh</i>	<i>17</i>
2.2.9 <i>Finance Manager - Emmanuel Olutayo.....</i>	<i>19</i>
2.2.10 <i>Assistant Finance Manager - Penny Nicole.....</i>	<i>20</i>
3.0 Deliverables.....	21
4.0 Project Management Methodology.....	23
4.1 <i>Initial Requirements Capture and Specification</i>	<i>24</i>
4.2 <i>Project Plan</i>	<i>25</i>
4.3 <i>Iterative Process</i>	<i>25</i>
4.3.1 <i>Select User Stories</i>	<i>25</i>
4.3.2 <i>Design and Planning Stage.....</i>	<i>25</i>
4.3.3 <i>Implementation and Testing Stage.....</i>	<i>26</i>
4.3.4 <i>Verification Stage.....</i>	<i>27</i>
4.4 <i>Final product check</i>	<i>27</i>
4.5 <i>Quality Assurance Reviews</i>	<i>28</i>
5.0 References	29
6.0 Appendix A - Documentation Templates	30
6.1 <i>General Document</i>	<i>30</i>
6.2 <i>Standard Document Cover Pages.....</i>	<i>31</i>
6.3 <i>Standard Document Example</i>	<i>34</i>
6.4 <i>Meeting Minutes Template</i>	<i>35</i>
6.5 <i>Weekly Payroll Template.....</i>	<i>36</i>
6.6 <i>Review Document Template.....</i>	<i>37</i>
6.7 <i>Test report template.....</i>	<i>38</i>
6.8 <i>Market Research Surveys</i>	<i>39</i>
6.9 <i>Weekly Financial Review Template</i>	<i>41</i>
6.10 <i>Time sheets Template.....</i>	<i>42</i>
6.11 <i>GANTT chart.....</i>	<i>43</i>
7.0 Appendix B.....	44
7.1 <i>Pseudo Code Example</i>	<i>44</i>
7.2 <i>Java Coding Standard Example.....</i>	<i>44</i>
7.3 <i>UML Class Diagrams</i>	<i>46</i>
7.4 <i>Initial GUI wireframe</i>	<i>46</i>
7.4.1 <i>Home page wireframe.....</i>	<i>46</i>
7.4.2 <i>Media Player wireframe</i>	<i>47</i>

Quality Assurance Manual

Work produced by:

SWEng Group 4

Company Name:

sofia

Members:

Calum Armstrong (ca718)
Daniel Berhe (dbb503)
Alexander Cash (ac1054)
Samuel Hall (sh1157)
Alistair Jewers (amj523)
Penelope Nicole (pen501)
Emmanuel Olutayo (epo500)
Samuel Raeburn (sr896)
Jake Ransom (jr829)
Lewis Thresh (lt669)

Department of Electronics

University of York



1.0 Introduction

1.1 Company Profile

The company was created as a software development organisation with the intention of creating teaching software. Our team is comprised of ten university students all working towards master's degrees in electronic engineering, all capable of innovating and collaborating in a dynamic manner to create high quality products.

Our product design ethos is customer-centric which enables us to produce products which provide real value and usability. We aim to embrace and further introduce technology into education which will encourage teachers and students alike to re-evaluate where learning happens.

1.2 Vision

- “To provide teachers and students with the facility to extend their learning outside of the classroom”
- “To create an intuitive and easy-to-use product, to get the user up and running as quickly as possible”
- “To make a product with real scope for further extension and development, to give customers new features they want at later dates”
- “To reduce the paper trail usually associated with teaching, thus saving trees”
- “To help people learn how and when they want to”

2.0 Jobs and Responsibilities

2.1 Organisation structures

For an organisation to achieve success there must be a clear and understandable vision or objectives which must be agreed upon by the organisation. Therefore for these objectives to be met, each group member must have a responsibility and a vital role to play to attain the set vision of the organisation. Hence, why roles have been given to each person in this company see Figure [1] for the roles given to each individual in the group and how the company is structured.

The project manager has the responsibility of ensuring that everyone in the group is pulling their weight appropriately, he also brings the group together as an effective communicator sets up group meetings, makes sure that every department in the group is working to its fullest potential and he is the first point of contact whenever any department have any problems. He is the person that liaises with the customer and also makes sure the company's vision and objectives are followed to the letter.

The figure below shows the hierarchy that has been agreed upon by the whole company. These roles in relation to each other are organised just for clarity sake and do not imply that anyone is superior to another. Communication between all group members is done mainly during weekly meetings set up by the project manager for planning and progress reports. Also after every objective is met there is a meeting to critically evaluate it and the group gets to scrutinise how genuine and how fit for purpose it is.

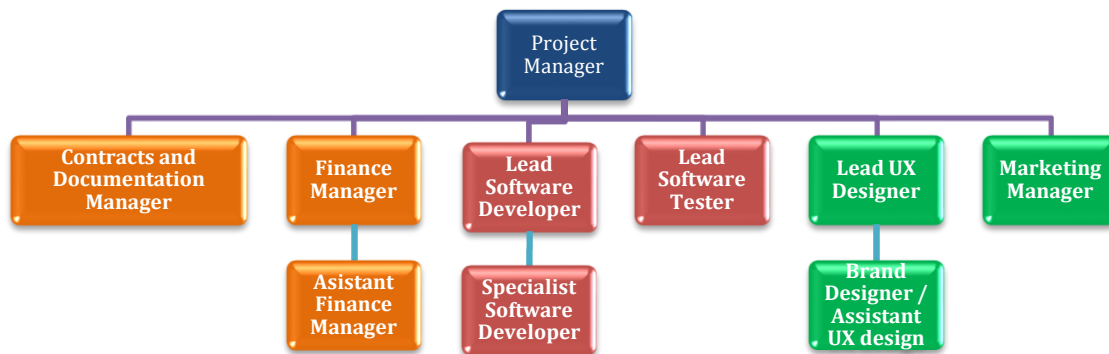


Figure 1- Diagram to show Companies hierarchical Structure

Orange – Contracts and admin team

Red – Software team

Green – Product design, marketing and branding

2.2 Job Roles

2.2.1 Project Manager – Alex Cash

2.2.1.1 Role description

The Project Manager's primary objective is to ensure that the project is completed smoothly and efficiently by ensuring team members know what their tasks should be, and are involved and invested in the completion of the project. The role of Project Manager is heavily organisational and administrative, ensuring the team is focussed to enable progress. Some of the tasks that are completed by a Project Manager are:

- Ensuring the group is made aware of all necessary information, such as meeting times, by email, social media, and in person
- Organising frequent meetings between the entire group
- Ensuring team members are kept up to date with information and tasks when they are unable to attend group meetings
- Making sure team members know where we are as a team, in terms of our iterative cycle process
- Ensure the team is focussed on the iteration at hand, so as to increase efficiency
- Communicating with the client/customer whenever necessary and acting as the first point of contact should the client/customer need to contact us. This requires frequent emails and presence at all meetings
- To take the lead in running group meetings ensuring conversation remains on topic, minutes are being taken, and that points on the agenda for that meeting are being discussed and covered suitably
- Collaborate with other team members to decide on the direction for the product, in terms of both design and user experience
- Lead the planning effort for each iteration required for the completion of the product, organising planning meetings and making planning decisions where necessary, as long as overseeing test efforts
- To plan the time requirements for the project and work with the finance managers to assess what our costs will be

- Work with others to assign tasks to the relevant team members. For example, assigning particular development tasks to the development managers for them to delegate where necessary
- Organise infrastructure for services such as task tracking and code versioning. This also comes with the associated task of managing these services once in use and maintaining them for utilisation by the team
- Ensure reports and documentation are completed correctly and on schedule

2.2.1.2 Risk Management

Risk	Risk Mitigation
Team members are finding themselves without anything to work on after completing a task.	Ensure planning is both thorough and enables flexibility, so that team members can always find a task to pick up that they would be suitable for completing. Encourage a working style that is dynamic so that team members can work with different people depending on what task they are completing, making it easier to reduce time spent waiting for new work with a specific partner.
Deadlines approaching and work is not at the necessary completion point.	Once again ensure there is thorough planning beforehand so that the team can work at full pace when approaching a deadline without needing to stop to recomplete the planning phase. Work to organise the team to ensure members are being utilised in the most efficient way; have our strongest developers working to develop important and difficult sections whilst our most testing-minded team members test the product.
Team member is unable to work due to illness/extended absence.	Re-organise the team in order to cover the work of the absent team member in order to pick up the slack and maintain almost 100% working pace.
The product has deviated from the original specification and planning	Adjust to these new working conditions. We are operating using an agile methodology, which enables us to change our direction easily. This may require meetings with the customer to ensure any changes are approved by them before being implemented, but should not hinder the product completion as a whole. Also, using the agile method means that there shouldn't be an abundance of prior product planning, so having to scrap some previous planning should not have lost the team too much time

2.2.1.3 QA Metrics

Metric	Measurement
Team is content with their work and feel confident they know what their tasks are.	Continue to organise regular group meetings but ask the group if they feel well organised, content, and know their tasks. If not, how it can be improved.
All deliverable deadlines met with work of a good standard.	Ensure all documents/deliverables are submitted as per the deadline, ensuring beforehand that the whole team is happy with the quality of the work that is being submitted.
Final product is at least representative of the initial plan (not required to be exactly the same) and is of a high standard.	After completion, compare the finished product with the initial product plan to see if we have met our initial requirements. Also, gather opinions of others (outside of the team) on whether the product is of a high standard.
Final product delivered on time.	Ensure product is in a finished stage when it is submitted to the customer on the deadline. This will be dependent on the team's confidence that the product has met our requirements.

2.2.2 Lead Software Developer - Alistair Jewers

2.2.2.1 Role Description

The Lead Software Developer is responsible for the technical design and implementation of the software. This includes the structural design and integration of the different software elements and design of the data path through the program, as well as coordinating the implementation efforts of the team. A full understanding of the project requirements and the associated functional specification is required to ensure the product developed meets the client's wishes. He/she should be in frequent communication with the Lead Software Tester to ensure the correct functionality of the software and to identify and correct bugs, as well as the project manager when coordinating coding efforts. Furthermore he/she should continually liaise with and oversee team members working on specific aspects of the software, to ensure seamless integration of the various software elements.

The Lead Software Developer tasks should include:

- Develop the software design at the beginning of each iteration based on the functional specification and the functionality chose for that iteration.
- Determine the methods and techniques to be used in the design step of each iteration in conjunction with the project manager.
- Identify the specifics of the coding process, including the language to be used, a specific coding standard and (in the case of code that is to be sold) an API documentation method.
- Identify the technical details of the design such as code structure and hierarchy, data path, encapsulation and object oriented techniques.

- Monitor and coordinate the implementation process during each iteration and the integration of the different code elements and program functionalities.
- Ensure that coding standards are adhered to across the whole code base and that there is consistency between all parts of the code.
- When necessary deliver concise training sessions and materials to enable team members to use coding techniques and tools they are not familiar with.

2.2.2.2 Risk Management

Risk	Risk Mitigation
Missing functionality.	Well defined set of functionalities chosen from the specification for each iteration and referred to regularly during implementation. Regular communication with Lead Software Tester to identify the full necessary set of test cases.
Incompatibility of different software elements.	Rigorous design details with emphasis on code hierarchy and structure as well as data flow established at the start of each iteration, with reference to all proceeding iterations. Integration testing performed early and often. Regular discussion and code review with the team members responsible for each software element.
Code failure and/or bugged code.	Regularly review code in conjunction with Lead Software Tester and the author of the code under review to identify failures and bugs. Cooperate with Project Manager to assign appropriate manpower to correcting issues, based on priority of the code element. Communicate with Project Manager over the use of version control software to manage code additions and issues.
Code inconsistency.	Decide on and provide team with a full coding standard. Review code regularly and monitor all additions to the code base.

Should they arise, these issues should be reported to the Project Manager in the first instance and then the relevant team members as soon as is feasible. The Lead Software Tester should also be regularly informed.

2.2.2.3 QA Metrics

Metric	Measurement
Features implemented vs features planned	Compare the list of finished and tested features to the original set of required and potential features. This comparison

	should be made during each iteration, for that iteration's feature set.
Time spent coding vs planned.	Track the start and end of the coding process during each iteration.
Bug tracking.	Use bug or issue tracking software to maintain a list of all bugs that have been identified and whether they have been fixed and tested. Bugs should be assigned to relevant people for fixing.

2.2.3 Lead Software Tester - Sam Raeburn

2.2.3.1 Role Description

The main role of the Lead Software Tester is to oversee and manage the testing (and procedures of testing) of software produced by S O f i a. In order to complete their responsibilities successfully they will require a clear and intricate understanding of how the finished product will work, including, but not limited to; how different classes interact, the changes required between iterations as well as individual functions. They are required to be able to view the project as a completed product at all times and should always be considering the integration of software. The Lead Software Tester is required to be able to switch mentally between viewpoints of the product, from the development side to the client side, this is vital in ensuring that time is not wasted producing unnecessary or incorrect software. Testing GANTT charts should be produced to guide the testing and should be adhered to at all times, an example of a GANTT chart is shown in Appendix A. Test reports should be created by the Lead Software Tester and completed by any member of staff involved with testing. An example of a testing sheet is shown in Appendix A. The bulk of the Lead Software Testers' work will come during the last two stages of each iteration, the implementation and testing stage and the verification stage. The fundamental responsibilities of the Lead Software Tester are:

- Outline and manage testing procedures.
- Create an appropriate testing plan.
- Represent and schedule testing plan using a GANTT chart.
- Assign tasks associated with testing to team members with accordance to testing plan.
- Collect test results from team members by means test reports
- Produce reports summarising the testing stages.

2.2.3.2 Risk Management

Risk	Risk Mitigation
Issue occurs in software but it is not	Arrange a meeting with the Lead Software

immediately clear what is causing issue.	Developer and team member responsible for particular piece of software to locate problem.
Software produced to out-dated specifications.	Ensure every team member involved with development communicates appropriately and always has access to the latest version of the software from the GitHub repository before beginning any coding.
Issues when integrating one or more modules of software.	Communicate with the Lead Software Developer to resolve issue.
Testing or integration deadline not met.	Consistently check GANTT chart to try and avoid this, if it does occur reschedule testing or integration and, if necessary, subsequent tasks.

2.2.3.3 QA Metrics

Metric	Measurement
Features implemented vs features tested.	Compare list of features with list of features tested obtained from test reports.
Software modules vs test reports.	Compare the amount of modules created to the amount of distinct test reports.
Changes due to bugs vs test failures	Failures in testing should result in a change in code.

2.2.4 Contracts and Documentation Manager - Calum Armstrong

2.2.4.1 Role Description

Given the time restraints often associated with projects, it is not always possible to internally write all code required for each piece of software. It may therefore be necessary to buy in code from other companies working on similar projects which would require contracts to be drawn up in order to secure a reliable exchange. Likewise, code can also be sold to generate extra income. The Contracts Manager should take responsibility for these sales, negotiating terms on behalf of the company and liaising with external contractual representatives.

Throughout software development many different documents will be produced by each employee at the company. It is important these documents are suitably filed and available to access by others. The Documents Manager should take responsibility for organizing all documentation produced and ensuring that it is accessible to whoever requires it. When made publically available, they should also be archiving old versions of documents and make these easily accessible. Minutes should be taken for every meeting and suitably archived to monitor progress. These should ideally be taken by the documents manager, however if this is not possible, a substitute should be made. Responsibility should also be taken for ensuring employees know what they may be expected to do as a result of any documentation produced, for example, actions to be taken after a meeting.

Tasks for the Contracts and Documentation Manager include:

- Take minutes at each company meeting and provide actions for employees and a result
- keep track of the progress made on each action
- Keep an up to date register of meeting attendance
- Organize any documentation produced throughout the project
- Ensure any documentation complies to company standards / templates
- Ensure all documentation is up to date
- Archive all documentation in a company-wide accessible format
- Carry out regular reviews of all documentation to maintain standards
- Request and verify time sheets from employees on a weekly basis and update payroll
- Take a lead role in the exchanging of contracts with other companies
- Provide a point of contact for other companies to approach
- Keep the Project Manager and employees informed of any offers / negotiations made by other companies
- Represent the company during contract negotiations
- Work with the marketing team to secure the sale of our product

2.2.4.2 Risk Management

Risk	Risk Mitigation
Missing / corrupt document	Keep a backup of all documentation produced, even if updated versions are available
Company target not met	Keep detailed minutes of meetings in order to review target proposal and employee responsibilities regarding the target
Incompatibility with code purchased from external company	Include detailed compatibility clauses within purchase / licensing contract

2.2.4.3 QA Metrics

Metric	Measurement
Contractual errors	Number of amendments made to contracts, or employee dissatisfaction as a result of a missing clause causing problems in code integration / implementation
Documentation availability	Number of requests for documents / updated documents not made publically available
Timesheet management	Number of timesheet summaries provided to finance by mid-day Tuesday

Metric	Measurement
Document archiving	Number documents that became corrupted / were lost and were unrecoverable due to a lack of backups

2.2.5 Specialist Software Developer - Daniel Berhe

2.2.5.1 Role Description

The Specialist Software Developer will play a role in overseeing software coding and will ensure consistent coding styles are followed. They will be responsible for coordinating the coding efforts on a specific part of the software while keeping regular contact with the Lead Software Developer, Lead Software Tester and Software Development Team. He/she will have full understanding of the product specifications, target consumer needs and company software development standards. Tasks of the specialist software developer will include:

- Ensuring consistent coding styles agreed with the lead software developer are followed.
- Help monitor and coordinate the implementation process alongside the Lead Software Developer.
- Help decide on design and development methods.
- Co-monitor the coding process.
- Ensuring software is developed within specified time frame.
- Ensuring that software is developed up to the standards of the company.
- Help in ensuring high level and module designs are as detailed as possible

2.2.5.2 Risk Management

Risk	Risk Mitigation
Loss of code	Using version control software to ensure all codes are backed-up.
Delay of software development	Assign more developers to the task at hand.
Unclear specification	Organise a meeting involving all company personnel to discuss the requirement further. If necessary, contact customer to ascertain a clear specification.
Late requirement changes	The latest requirement can be integrated in the planning of the next iteration
Code inconsistency.	Ensure coding styles agreed beforehand are followed by all software developers.

2.2.5.3 QA Metrics

Metric	Measurement
Features implemented vs features planned for specific sections of the program	Compare the list of finished and tested features to the planned ones.
Bug tracking of specific sections of the program	Record the number of bugs found during testing of the modules by the testing team.
Comments in code.	Comparison of lines of comments to lines of code. Documentation comments per class.

2.2.6 Marketing Manager - Jake Ransom

2.2.6.1 Role Description

In designing and implementing a product, it is important to be aware of what the current demand is for any given idea and then separating it from any competition. Therefore the main role of marketing will be to promote the products awareness, features and sales to potential custom. Key responsibilities include:

- Research new product opportunities and demand for said product
- Ensure the product is appropriate for target audience
- Analyse future customer needs and requirements
- Launch Campaigns
- Liaising with the design team for quality branding and representation of customer needs
- Design presentations to effectively market and sell the product
- Research Current Competitors
- Use market research and competitor information to price products

2.2.6.2 Risk Management

Risk	Risk Mitigation
Competition	Produce high quality product, high market awareness of brand and product, give quality after sales services and provide unique features
Pricing (Over pricing, insufficient funds)	Research similar product prices to pitch against, pitch and sell product code to generate income
Target Audience changes	Early research and evaluation for end user requirements, make allowances for product to appeal to several target audiences
Market Access	Make sure that product keeps to product certification requirements,

2.2.6.3 QA Metrics

Metric	Measurement
Brand Awareness	Polls such as surveys
Customer Satisfaction	Customer feedback through reviews
Market Share	Market Research
Sales	Amount of units sold

2.2.7 Lead User Experience Designer - Sam Hall

2.2.7.1 Role Description

The product should show a clear and consistent style to the user. To that end, a UX (User Experience) Design should be created. This encompasses elements such as the Interface, any graphics and colour.

The UX governs the user's first impression, therefore it is important to present the product in the best way possible and ensure design continuity throughout the product. A portion of the company's budget will go into research and development of the UX and collaborations with the brand design team are essential to ensure we are all moving in the same direction.

Specific responsibilities include:

- Liaise with brand design team to decide application theme
- Research modern application UX design
- Mock up designs and take feedback from group members
- Liaise with Lead Developer to ensure designs can be implemented
- Creating the UX for the coding team to work towards

2.2.7.2 Risk Management

Risk	Risk Mitigation
Delay in design completion	Ensure slack time is available for anyone needing to start the next stage of the project. Alternatively – Re allocate tasks in order to complete the design efficiently

Insufficient Budget	Calculate maximum potential budget and ensure there is a minimal cost route that can be taken to ensure the completion of an acceptable design.
Inability to continue design throughout the application due to it being inappropriate	Plan all aspects of the application and identify where appropriate continuity of design exists.
Inconsistent design	Keep liaising with brand design team, contract, and documentation teams to ensure that design continuity from company documentation through to application is met.
Poor Design	Ensure enough research is done to limit the possibility of a poor design
Design cannot be coded	Arrange regular meetings with Lead Developer to check possibilities

2.2.7.3 QA Metrics

Metric	Measurement
Icons	Check that any icons required by the pre-existing design are completed on time and are of high quality
Inconsistent design	Keep liaising with Assistant UX Design, contract, and documentation teams to ensure that design continuity from company documentation through to application is met.
Appropriateness	Ensure design meets spec as defined in functional spec. The design should allow for all the features to be implemented with no design done by the coding team.
HTML Website/Tour	Websites are completed on time, are easy to navigate and show all the required information

2.2.8 Company & Application Brand Designer / Assistant User Experience Designer - Lewis Thresh

2.2.8.1 Role Description

To design a product with requirements specified by a user, the user must be able to deal and communicate with the group in its entirety. This is made possible by banding together the involved work force through the medium of a company.

The company branding can influence the users first impression, therefore it is important to present the company in the most appropriate way and to then ensure design continuity throughout the company's products. To ensure this, a portion of the company's budget will go into research and development of the company logo and collaboration with the Lead

UX Designer is essential. A portion will also go into the development of the application theme and development.

As assistant user experience designer, I will assist in the design of the application user interface, provide feedback to prototype versions, and develop the final user interfaces, ensuring they appropriately represent our company.

Specific responsibilities include:

- Liaise with Lead UX Designer to decide application theme
- Prototype designs and take feedback from group members
- Document logo versions
- Design final company and application logos
- Assist members of the team where needed
- Develop application GUI's

2.2.8.2 Risk Management

Risk	Risk Mitigation
Delay in design completion	Ensure slack time is available for anyone needing to start the next stage of the project. Alternatively – Re allocate tasks in order to complete the design efficiently
Insufficient Budget	Calculate maximum potential budget and ensure there is a minimal cost route that can be taken to ensure the completion of an acceptable design.
Inability to continue design throughout the application due to it being inappropriate	Plan all aspects of the application and identify where appropriate continuity of design exists.
Inconsistent design	Keep liaising with Lead UX Design, contract, and documentation teams to ensure that design continuity from company documentation through to application is met.
Impractical Design	Liaise with Lead Developer to ensure all designs are feasible when it comes to development.
Loss of designs/code	Use version control to ensure all designs and code are backed up

2.2.8.3 QA Metrics

Metric	Measurement
Design cost	Hours spent per individual researching and producing logo and design.
Deliverables	Number of logo design iterations complete. Number of GUI iterations
Appropriateness	How close do the initial specifications match the final product
Defects	Number of elements that do not meet design specification

Assistance	Hours spent assisting other department members
------------	--

2.2.9 Finance Manager - Emmanuel Olutayo

2.2.9.1 Role Description:

In a company, one of the most important aspects are the financial activities; therefore a finance team is put together to take on this complexity. The finance team will have the responsibility of: managing the cash flow, book keeping, creating financial documents such as balance sheets and profit and loss documents, providing the Project Manager with the essential financial information concerning the financial health of the company, management of wages, making sure the company does not go bankrupt, and many other duties. Concerning the Financial Manager, their roles will include:

- Management of the budget.
- Financial projections
- Pricing of the product
- Critical analysis of where money should be invested
- Looking into investment opportunities
- Deal with queries from the bank
- Liaise with project manager and financial backer
- Conferring with members of the company concerning financial decisions that are made
- Conduct weekly reviews with the finance team to look into opportunities to reduce costs
- Stay informed concerning changes in monetary regulations and enactment.
- Analyse market trends
- Create good relationships with external contacts including financial backer
- Risk mitigation

2.2.9.2 Risk Management:

The fear of financial risk haunts every business. This means that risk must be considered and mitigated for before they appear.

Risk	Risk Mitigation
Misallocation of funds	Make sure all investments are legitimate and all hours that are logged in by each member of the team are appropriate.
Credit risk	Make sure all contracts are analysed by the project manager, finance team and contract and documentation manager to make sure that all the terms and conditions of the financial backer can be met.
Risk of Bankruptcy	Make sure finances are checked every week and adapt as required by speaking to the group at the beginning of every week on the financial state of the company. Make sure we optimise how much to get from financial backer.

Poor budgeting	Ask company employees for estimates on labour hours and factor that in to the budget.
----------------	---

When or if any of these problems occur, the first point of call would be the Project Manager, then the financial backer or bank.

2.2.9.3 QA Metrics:

Metric	Measurement
Return on investment	Check returns with appropriate department in company
Pay Back period	Pay interest promptly
Operating expense control	Check weekly accounts against what has been predicted
Return on assets	Make sure the most suitable group member is doing what is required of them as efficiently as possible therefore saving money.
Legitimacy of Financial Documents	Check cash flow every month or week to make sure all cash inflow in financial documents add up

2.2.10 Assistant Finance Manager - Penny Nicole

2.2.10.1 Role Description

In a company, one of the most important aspects is the financial activities; therefore a finance team is put together to take on this complexity. The Assistant Finance Manager aids and collaborates with the Finance Manager – they are essentially a supportive role for the Finance Manager. Concerning the Assistant Finance Manager, their roles will include:

- Produce up to date weekly accounts and suggest possible reduction opportunities as appropriate alongside Finance Manager.
- Compile and submit financial documents (eg. business plans, financial reports) on time to be delivered to the appropriate individual.
- Manage loan and interest repayments promptly.
- Manage other payments promptly (eg. wage, rent, utilities, IT infrastructure).
- Pricing of the product.
- Conferring with members of the company concerning financial decisions that are made.
- Avoidance of bankruptcy.

2.2.10.2 Risk Management

Risk	Risk Mitigation
Misallocation of funds	Make sure all investments are legitimate and all hours that are logged in by each member of the team are appropriate.

Credit risk	Make sure all contracts are analysed by the Project Manager, finance team and Contract and Documents Manager to make sure that all the terms and conditions of the financial backer can be met.
Risks of Bankruptcy	Make sure finances are checked every week and adapt as required by speaking to the group at the beginning of every week on the financial state of the company. Make sure we optimise how much to get from financial backer.
Poor budgeting	Ask company employees for estimates on labour hours and factor that in to the budget.

When any of these problems occur, inform the Finance Manager.

2.2.10.3 QA Metrics

Metric	Measurement
Accuracy of financial information	Check financial documents with the Finance Manager, and other employees with financial knowledge. Produce weekly financial reports, and use them to check consistency of financial information in related documents, including the Payroll and the Financial Reports.
Pay Back period	Pay interest promptly.
Operating expense control	Check weekly accounts against what has been predicted in the Financial Reports.
Financial Prediction	Produce financial plans regarding the development of the initial product in the first six months of business.

3.0 Deliverables

In addition to this, all Documents should be uploaded to the company's online document repository to be archived by the Contracts and Documents Manager

Deliverable	Author	Recipient	Deadline
Recurrent Deliverables			
Timesheet	All Employees	Contracts and Documents Manager	Weekly (Monday)
Timesheet summaries	Contracts and Documents Manager	Finance Manager	Weekly (Tuesday)
Weekly Financial Statement	Finance Department	Project Manager	Weekly (Wednesday)
Minutes	Contracts and Documents Manager	All Employees	After each meeting

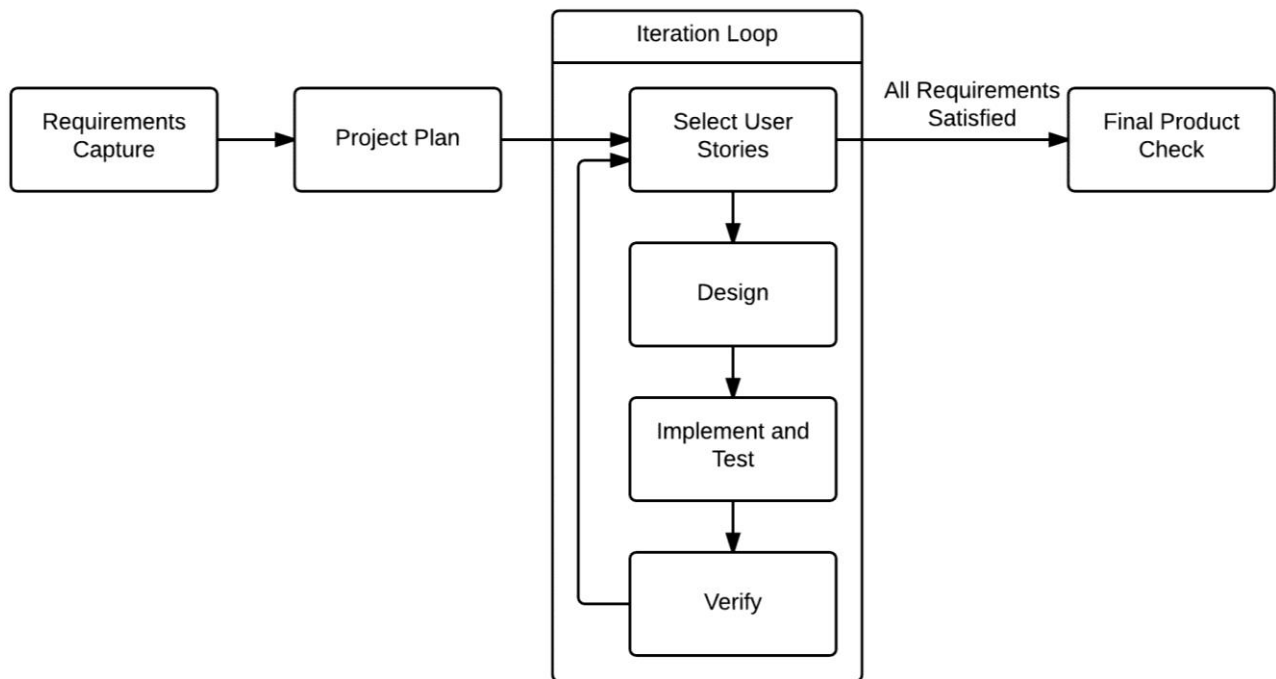
Iteration Plan	Lead Software Developer	All Employees	Start of each new iteration
Project code	All Employees	Lead Software Developer, Specialist Software Developer	Throughout each iteration
Code Quality Assurance Reviews	All Employees	Lead Software Developer, Contracts and Documents Manager	For each new segment of code
Test Reports	Lead Software Tester	Lead Software Developer, Specialist Software Developer	For each test carried out on iteration
External Deliverable Quality Assurance Reviews	All Employees	Project Manager, Contracts and Documents Manager	For each deliverable produced for external deliverable
Time variable Deliverables			
Contracts to Buy / Sell Code	Contracts and Documents Manager	Project Manager, Contracts and Documents Manager	Written up when agreed with other party
Product Specification	Project Manager	Client, All Employees	End of Initial requirements capture
Project Plan	Project Manager	All Employees	End of Initial requirements capture
Initial UI Design	Lead UX Designer	Lead Software Developer	End of Initial requirements capture
Market research summary	Marketing Manager	Project Manager, Lead Software Developer	End of Project Plan
UI Design	Lead UX Designer	Lead Software Developer	End of Market Research
Brand / Company Image Design	Brand Manager	Project Manager, All Employees	End of Market Research
Fixed Deadline Deliverables			
Functional Specification	Project Manager	Client, All Employees	Friday 30 th January 2015
QA Manual	Brand Manager	All Employees	Monday 2 nd February 2015
Financial Business Plan	Finance Department	Project Manager, Client	Friday 6 th February 2015
Project Wide Standards Document	All Employees	Project Manager, Contracts and Documents Manager	Thursday 12 th February 2015

Tender presentation	Marketing Manager	Project Manager, Client	Monday 16 th February 2015
Financial Report I	Finance Department	Project Manager, Client	Friday 20 th February 2015
All Contracts written up	Contracts and Documents Manager	Project Manager, Contracts and Documents Manager	Tuesday 24 th February 2015
Financial Report II	Finance Department	Project Manager, Client	Friday 6 th March 2015
Final Product test plan	Lead Software Tester	Lead Software Developer, All Employees	Friday 13 th March 2015
Financial Report III	Finance Department	Project Manager, Client	Friday 1 st May 2015
Financial Summary Report	Finance Department	Project Manager, Client	Friday 22 nd May 2015
Demonstration and Sales Presentation	Marketing Manager	Client	Monday 1 st June 2015
Finished Program	Lead Software Developer, All Employees	Project Manager, Client	Thursday 4 th June 2015

Figure 2 – Deliverables and reports expected throughout project

4.0 Project Management Methodology

The following diagram shows how we will be working overall i.e the agile flow from start to finish. Each section is explained in more detail below.



4.1 Initial Requirements Capture and Specification

At the start of a project the customer will specify what product they would like developed and what the product should do. These requirements will most likely be provided in descriptive, non-technical 'user stories' which identify the necessary features of the software, but not how these features will be achieved. The company, led by the Project Manager, will then analyse the client's requirements to produce a detailed initial specification for the project. The client should be involved in this process to identify and discuss any inaccuracies between the specification and client requirements. The specification created is then referenced at the start of each iteration, when specific features are being chosen for that iteration. The specification is also updated if necessary after each iteration as the project evolves. The customer should be involved in this process.

This process is summed up in the following steps:

- A list of client requirements is provided by the customer, who states what the product must do in general, non-technical language.
- The team as a whole, led by the Project Manager, will analyse these requirements with input from each department with regards to the feasibility of each requirement.
- A product specification will be derived from this analysis along with an initial budget estimate from the financial department. The product specification will outline the functionality of the software to be developed. This process will also identify any inaccurate or inconsistent client requirements.
- The Project Manager will liaise with the customer with regards to any potential changes to the requirements.
- Upon agreement of the final requirements the Contracts and Documentation Manager will produce the final specification document. The Project Manager can then use this to produce a project schedule in the form of a Gantt Chart 7.0 shown in Appendix B. This will include approximate timings for each iteration.

- The finalised specification document can then be distributed to the team by the Contracts and Documentation Manager, beginning the design phase of the project.
- This specification document should be reviewed after each iteration, as necessary changes may become apparent as the project develops.

4.2 Project Plan

The Quality Assurance Project Plan is an essential part of the overall Quality Assurance system. It will outline all major deliverables and deadlines that the project must adhere to and follow. The core concept of the plan is to reassess requirements as the project progresses. This will concentrate efforts on producing a higher quality of attainable deliverables and improve consistency throughout.

Using client requirements some preliminary market research should be done to assess similar products to determine extra functionality that would benefit the user. It will give an indication to the size of the target audience and establish key performance indicators; allowing the product to be benchmarked against the current marketplace.

The project will use agile methodology to build a basic timeline structure, where only major key stages will affect how long a set of tasks and processes should take. Iterations will not follow a strict time length but instead be revaluated as progress is made by the relevant developers. For planning purposes iterations will be allocated time in the GANTT chart.

User stories should be devised from the client requirement covering all basic functionality needed. Selected user stories will enter into the iterative process as described in section 4.3 based on which task needs completing. Individual developers should provide an estimate of the time it will take to complete each iteration, enabling progress to be tracked effectively. This will provide a greater understanding of resource availability, which in turn improves flexibility to plan future iterations. Refining the resource planning will ensure teams are not overloaded which if not avoided will lead to lapses in quality.

4.3 Iterative Process

Once the initial specification has been confirmed the next step is to begin the iterations as per the Agile process. This involves repeatedly completing a number of stages to incrementally implement functionality into the program in order to satisfy the specification and user requirements. Each repetition of these stages forms one iteration and the stages that form a single iteration are outlined below.

4.3.1 Select User Stories

The Project Manager, in conjunction with the Lead Software Developer and Lead Software Tester, will select a specific set of functionalities to implement in the current iteration. These functionalities will satisfy one or more user requirements. This determines the work to be done during this iteration, and informs the design and planning stage. The functionalities to be implemented in each iteration may be selected in advance by the Project Manager during the project planning period, to better manage the team's use of time, however the choices will always be reviewed at this stage to allow the development plan to adapt and accommodate necessary changes.

4.3.2 Design and Planning Stage

The design and planning stage involves defining all of the tasks required to implement the functionalities chosen in the previous stage and satisfy the relevant user requirements. Once the tasks have been defined, a simple work breakdown structure will be created. The tasks will then be assigned throughout the development process by the relevant managers and tracked using our task tracking and minutes systems. Where necessary during this

stage the Lead Software Developer will produce pseudo code and/or UML class diagrams, examples of which can be seen in Appendix B. These should inform and aid the implementation and testing stage, but should not restrict the development if changes need to be made or a different approach adopted.

Specific software tools used by the company include:

- The Java programming language for software development.
- The Eclipse IDE for the production of Java code.
- The Microsoft Project planning software to produce GANTT charts.
- The Microsoft Excel spreadsheet software.

4.3.3 Implementation and Testing Stage

The implementation and testing stage is critical in ensuring the product works as desired. Following the agile methodology, code will be implemented and tested in parallel, to allow an adaptive approach to coding which can respond quickly to errors and bugs. The Lead Software Developer is responsible for co-ordinating the coding efforts of the team by distributing the coding tasks identified in the previous stage, and ensuring the development leads to the implementation of the functionality identified in the first stage. He/she will also communicate regularly with the Project Manager about the state of the code, and oversee the use of the version control system in order to maintain stability in the codebase and keep a record of the changes made. Simultaneously, the Lead Software Tester is responsible for co-ordinating the testing of this iteration's functionality whilst it is being developed. He/she is also responsible for outlining the procedures used in the testing process. Ideally, the testing of any particular module of software should be carried out by somebody who was not involved in the creation of that module. The Lead Software Tester will create test reports to be completed by any member of the team who is involved in the testing of a module; the report will contain details of what module is being tested and how. These test reports will help keep track of the project as well as keeping track of QA metrics. An example of a test report can be seen in Appendix A.

Any problems that arise either during the implementation or testing of the code that require adjustments to the schedule will be communicated to the Project Manager as well as between the Lead Software Developer and Lead Software Tester, and adjustments to the schedule and plan for this iteration will be made accordingly and the team informed.

Once any feature has been implemented, its functionality will be tested and any issues or bugs found recorded and communicated back to the software development team. All bugs found should be entered into the bug tracking system (GitHub Issues) and tagged with the appropriate information. The Lead Software Developer will then coordinate fixes. In this way the implementation and testing can be done in parallel, allowing a much faster and more flexible response to issues. Both automatic and manual testing methods will be used. A test early and test often ideology will be used rather than fully test driven development. The following rules apply during this stage:

- Any team member involved with the implementation and development of any code should adhere to the coding standard set by the Lead Software Developer. An example of this standard is shown in Appendix B.
- Any team member involved with the implementation and development of any code should make use of the version control system, and all code committed to the

system should include a comment describing the changes made and signed with the team member's name and email.

- Any team member involved with testing during this stage should complete testing reports for every module of software which they test, recording any errors or bugs, these should then be returned to the Lead Software Tester (either digitally or a hard copy).
- Any team member involved with either testing or implementation during this stage should update the relevant issue being tracked on GitHub.

During this stage it may become apparent that some aspects of the specification or some of the user requirements are either not possible, or not feasible in the project. Should this occur these problems should be communicated to the Project Manager as quickly as possible, and from there the Project Manager can, if necessary, contact the client and agree on a change to the specification and/or requirements.

4.3.4 Verification Stage

Once the specified functionalities have been implemented, the verification stage will test them against the relevant user requirements to ensure the aims of the iteration have been achieved. Regression testing methods will be used alongside integration testing methods to ensure that the implementation of a new feature hasn't broken a previously implemented feature. Any problems that are identified at this point will be communicated to the Project Manager and dealt with before the start of the next iteration. Team members involved in the verification stage must still adhere to the testing procedures outlined in the previous stage, including producing test reports. The aims of the verification stage are:

- To ensure that the functionality implemented during this iteration satisfies the elements of the specification identified in the first stage, and that the chosen user requirements are met.
- To ensure that the implementation of new functionality has not broken previously implemented functionality.

At this stage the Lead Software Developer will document any outcomes of the iteration that will affect future iterations, and update any relevant documentation such as UML class diagrams to reflect changes.

4.4 Final product check

After all iterations have been completed and all user stories have been achieved, the product will run through several manual checks to ensure the product is working overall as intended. The checked product will then be shown to the customer to ensure that it meets the customer's requirements. If not, then the Final Project Check iteration will be repeated until the customer is fully satisfied.

Once the customer is happy with the product, no more software development will be undertaken and the product will be formally pitched to the customer.

4.5 Quality Assurance Reviews

Delivering a high quality production is substantially influenced by the quality of the development process. The procedures followed during each iteration aim to meet these high quality standards. The main aims of the review session are to assess the quality of deliverables, the outcome of each iteration and quality of the code or to review completed documents. Passing the review is mandatory and is crucial in the progression to the next stage. Members of the review team will include:

- Deliverer: The person whose product is under inspection.
- Inspectors: Two members of the company will be involved in the reviewing. The inspectors will have full knowledge of the specification and the outcome of the iteration. The inspectors will be responsible for organising the review session, recording the outcome of the review as well as ensuring the company's review procedures are followed.

During the review, the inspectors will utilise documents such as: Test Report and Review document (see templates in Appendix A). In the case of code quality assurance review, the inspectors will comprise of lead software developer and lead testing manager. They will thoroughly inspect the code for quality, integrity, compatibility and consistency. In the case of external deliverable, like submitted documents, the inspectors will comprise of two company's employees who have relevant expertise. They will check the documents for quality, consistency, technical content and whether or not the document meets the company's layout standards. The outcome of the review will be recorded on the review document by one of the inspectors to submit to the Project Manager. Progression of the project is dependent on the approval of the project manager and inspectors.

5.0 References

- [1] - <http://www.gliffy.com/go/publish/image/4218693/L.png>
- [2] <http://prince2pm.files.wordpress.com/2011/12/gantt-chart.jpg>
- [3] <http://www.oracle.com/technetwork/java/codeconventions-150003.pdf>

6.0 Appendix A - Documentation Templates

6.1 General Document

All documents produced should be of a standardised format for consistency throughout the company. This should allow for easy comparison and collation throughout departments.

All documents will include a document control section. In the cases where documents must be altered after being originally published then this will be detailed in this section. Here the author of the change must be named, along with the date, and the summary of the change made. A reason for change may also be given at this point but is not required.

Documents should be formatted as follows:

- All documents should include a cover page, document control page, and contents page, as shown below. The formatting / Layout of these pages should not be changed and templates have been made accessible to all employees to ensure consistency.
- Font – Arial
- Main headings – Font size 14, Bold, Underlined
- Subheadings – Font size 12, Bold
- Body text – Font size 12
- Headings should be numbered numerically and hierarchically e.g.
 - 1.0 Main Heading*
 - 1.1 Sub heading*
 - 1.2 Second Sub heading*
 - 2.0 Second Main Heading*
- Dates should be of format – Day 00th Month Year
- All pages (apart from the cover page) should be numbered in the lower right hand corner. Page count should start from the cover page, but should not be shown on the cover page. i.e. the document control page should be numbered “2” and count from there.
- The date of most recent publication should be displayed in the upper right hand corner of each page (apart from the cover page)
- The Specific document title should be displayed in the upper left hand corner of each page (apart from the cover page)
- Table headings should be of font size 12, bold, and should have a grey background fill
- All figures should have a caption, in italics, detailing what the figure shows (this includes tables)
- Text should be fitted to page width

Examples of Documents are shown below. First the cover page, document control page and contents page, followed by specific document templates.

The only exception to these rules are the timesheets filled out by employees. These are simply single sheets of paper that are collated into a weekly payroll document and therefore do not require to be published as a full document themselves.

6.2 Standard Document Cover Pages

s o f i a

Cover Page

Specific Document Title

Document Control

Editor	Date	Update

Table of Contents

1.0 First Heading

	E
rror! Bookmark not defined.	
1.1 Sub Heading:	34

6.3 Standard Document Example

Standard Document Example

1.0 First Heading

1.1 Sub Heading:

Table		

Figure 1 – A table to demonstrate how a table should look

6.4 Meeting Minutes Template

Meeting Minutes

1.0 Attending

Name (Email) Job Title Yes/No

2.0 Update from previous meeting

Person Responsible	Action Carried Out	Complete?

Figure 1 – Update from previous meeting

3.0 Agenda

- Things to discuss

4.0 Minutes

4.1 Absences:

Person Absent	Reason?

Figure 2 – Reasons for absences

4.2 Previous Minutes Approved:

Date of minutes	Approved?

Figure 3 – Approval of previous minutes

4.3 Matters Arising:

Topic	Action	Person Responsible

Figure 4 – Topics discussed during meeting

5.0 Next Meeting

The next meeting has been arranged for:

Expected to attend:

6.5 Weekly Payroll Template

Weekly Payroll

1.0 (Date)

Employee	Job Role	Hours Worked	Pay
TOTAL			

Figure 1 – Payroll for week commencing (Date)

6.6Review Document Template

Review document

Project:

Report title:

Author:

Reviewed by:

Date: *Day 00th Month Year*

Metric	How measured	Produced by	Date	Results & Remarks

6.7 Test report template

Test Report

1.0 Test of GUI**1.1 Summary of testing****1.2 Prerequisites****1.3 Test importance****1.4 Test results**

Test Case	Expected Results	Obtained Results

1.5 Changes made due to test**1.6 Additional comments on test**

6.8 Market Research Surveys

Market Research Survey Template

*Project:**Report Title:**Circulation list:**Author:**Date:***Market Research Survey**

We would appreciate if you could find the time to complete this short survey and provide some feedback for our company, we value any input you provide.

Survey Introduction

(Short introduction to survey. E.g. we are designing a programme that will allow tutors to design online lessons for students....)

Personal Information (tick which is most applicable)

Gender:	Age Range:	Employment Status:	Do you have a disability:
<i>Male:</i> <input type="radio"/> <i>Female:</i> <input type="radio"/>	<i>16 – 21:</i> <input type="radio"/> <i>22 – 30:</i> <input type="radio"/> <i>31 – 45:</i> <input type="radio"/> <i>46 – 60:</i> <input type="radio"/> <i>60+ :</i> <input type="radio"/>	<i>Employed:</i> <input type="radio"/> <i>Un-employed:</i> <input type="radio"/> <i>Student:</i> <input type="radio"/> <i>Retired:</i> <input type="radio"/>	<i>Yes:</i> <input type="radio"/> <i>No:</i> <input type="radio"/>

Please list any Hobbies or interests you have (sports, cooking, music, etc.)*Please write here:*

.....

.....

.....

Please Rate the Questions from Very likely to Very Unlikely (tick applicable option)

<u>Questions</u>	<i>Very Likely</i>	<i>Likely</i>	<i>Possibly</i>	<i>Unlikely</i>	<i>Very Unlikely</i>
	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Please **Answer** questions with a short **sentence**

1.
.....

2.
.....

3.
.....

Additional Comments

Please write here:

.....
.....
.....
.....
.....
.....

Comments/discussion points/recommendations

6.9 Weekly Financial Review Template**Weekly Financial Review**

Week Starting :

Reviewed by :

£	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
Opening Bank Balance							
Income (eg. loan, contracts)							
Salaries							
Cost of Utilities							
Rent							
Cost of IT infrastructure							
Interest on loan							
Other expenditures (eg. contracts)							
Closing Bank Balance							

(Note: the Weekly Financial Review is kept for financial analysis ONLY and is not an official bank statement: for example, salaries are paid for the whole week the week after they are noted in the Weekly Financial Review, and “Bank Balance” does not refer to the actual bank balance. Please refer to the Financial Reports or Financial Performance Review and Profit and Loss Statement for the actual bank balance.)

Assistant Financial Manager Signature:**Documents Manager Signature :**

6.10 Time sheets Template

Timesheet

Employee Name: _____

Hourly Rate: £12.50

Week Commencing: _____

Day of Week	Date	Meeting Hours	Working Hours	Total Hours
Monday				
Tuesday				
Wednesday				
Thursday				
Friday				
Saturday				
Sunday				

Total Hours:			
---------------------	--	--	--

Employee Signature: _____

Secretary Signature: _____

6.11 GANTT chart

Gantt chart & Network Diagram Template

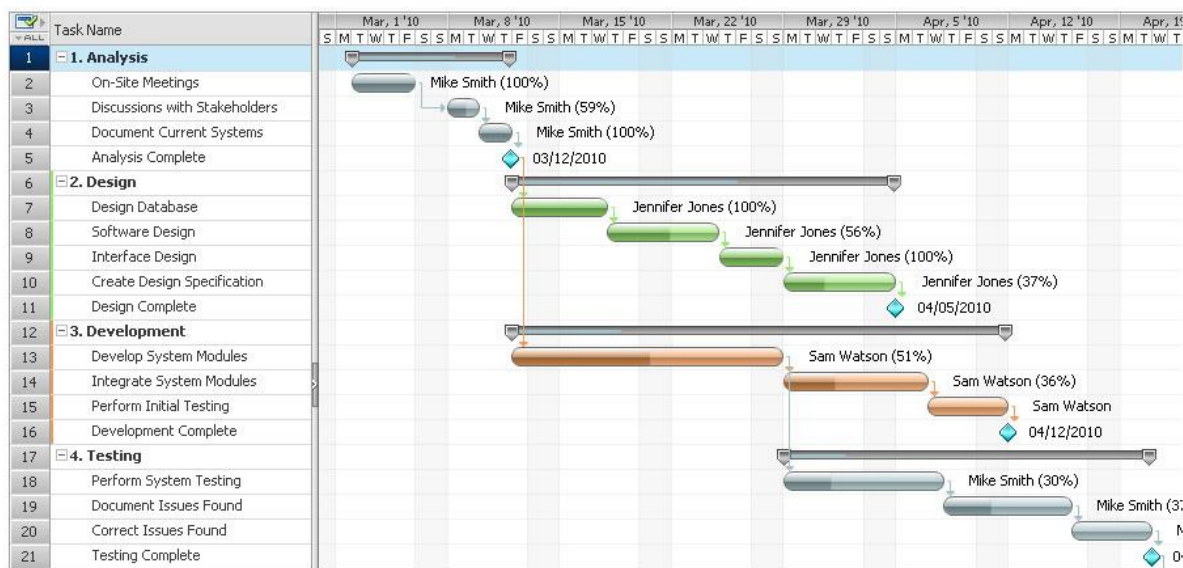
Project:

Report Title:

Circulation list:

Author:

Date:



Generic Gantt chart Fig [2]

Microsoft Project will be used to create the GANTT charts. The above template is a typical GANTT layout which will be used to show major deliverables, linked tasks, milestones, task durations and assigning resources.

Labels should be attached as shown with colours representing separate tasks and progression.

7.0 Appendix B

7.1 Pseudo Code Example

```

Method for drawing the objects {
    For each object in the object list {
        Get the object's coordinate position ()
        Get the object's size ()

        If object contains external media {
            Load the media ();
            Draw the object (position, size, media)
        } else {
            Draw the object (position, size)
        }
    }
}

```

7.2 Java Coding Standard Example

Full standard available at: [3]

```

/*
 * %W% %E% Firstname Lastname
 *
 * Copyright (c) 1993-1996 Sun Microsystems, Inc. All Rights Reserved.
 *
 * This software is the confidential and proprietary information of Sun
 * Microsystems, Inc. ("Confidential Information"). You shall not
 * disclose such Confidential Information and shall use it only in
 * accordance with the terms of the license agreement you entered into
 * with Sun.
 *
 * SUN MAKES NO REPRESENTATIONS OR WARRANTIES ABOUT THE SUITABILITY OF
 * THE SOFTWARE, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED
 * TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
 * PARTICULAR PURPOSE, OR NON-INFRINGEMENT. SUN SHALL NOT BE LIABLE FOR
 * ANY DAMAGES SUFFERED BY LICENSEE AS A RESULT OF USING, MODIFYING OR
 * DISTRIBUTING THIS SOFTWARE OR ITS DERIVATIVES.
 */

package java.blah;

import java.blah.blahdy.BlahBlah;

/**
 * Class description goes here.
 *
 * @version 1.10 04 Oct 1996
 * @author Firstname Lastname
 */
public class Blah extends SomeClass {
    /* A class implementation comment can go here. */

    /** classVar1 documentation comment */
    public static int classVar1;

    /**
     * classVar2 documentation comment that happens to be
     * more than one line long
     */
    private static Object classVar2;

```

```
/** instanceVar1 documentation comment */
public Object instanceVar1;

/** instanceVar2 documentation comment */
protected int instanceVar2;

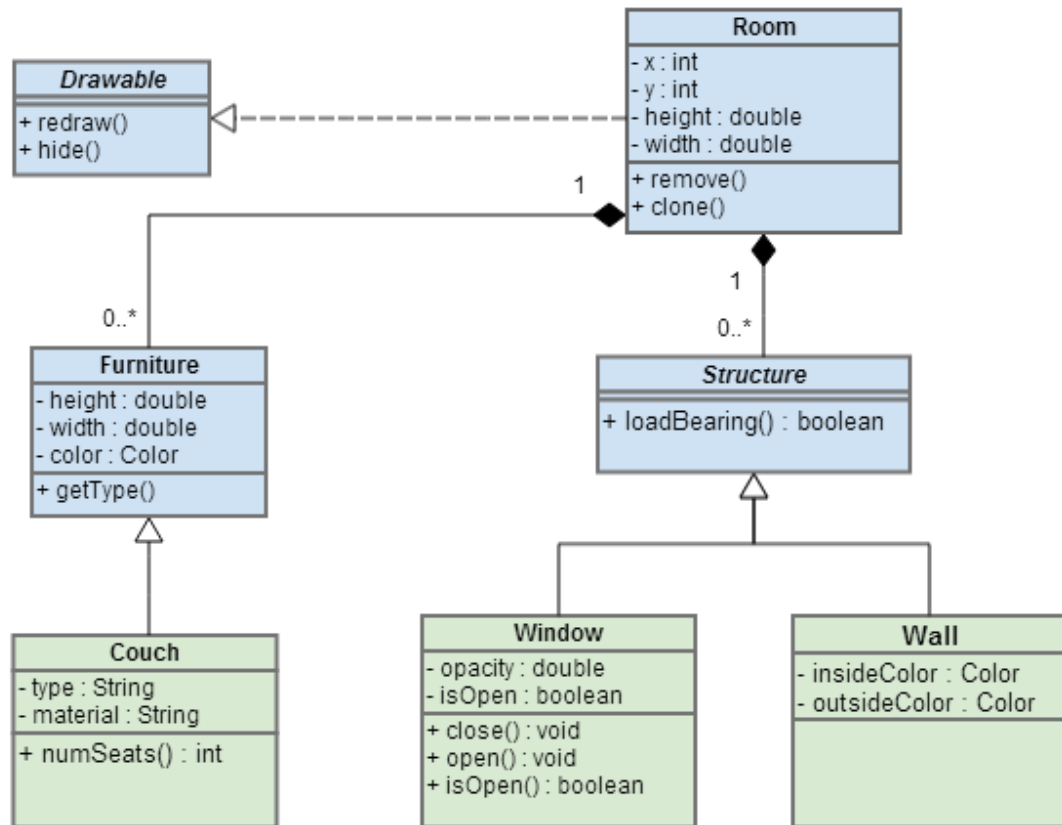
/** instanceVar3 documentation comment */
private Object[] instanceVar3;

/**
 * ...method Blah documentation comment...
 */
public Blah() {
    // ...implementation goes here...
}

/**
 * ...method doSomething documentation comment...
 */
public void doSomething() {
    // ...implementation goes here...
}

/**
 * ...method doSomethingElse documentation comment...
 * @param someParam description
 */
public void doSomethingElse(Object someParam) {
    // ...implementation goes here...
}
}
```

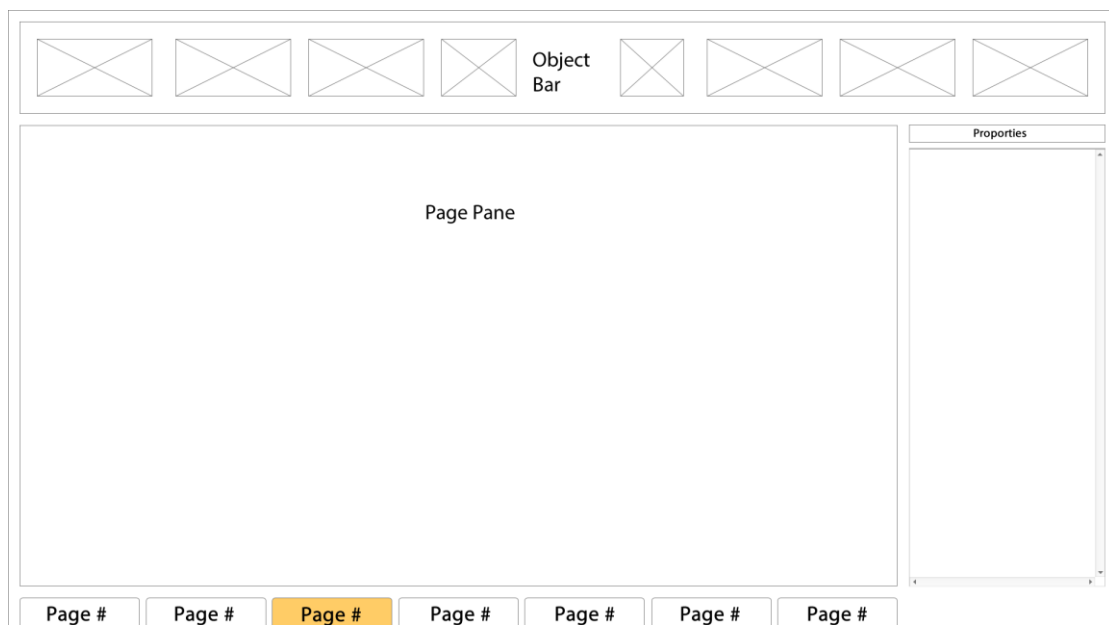
7.3 UML Class Diagrams



[1]

7.4 Initial GUI wireframe

7.4.1 Home page wireframe



7.4.2 Media Player wireframe

