**SOFTWARE SPECIFICATION**

1. **FEASIBILITY STUDY**

**Technical feasibility**

This project is to come up with a web-based application, movy app. The main technologies and tools that are required include:

* Html
* CSS
* JavaScript
* React
* Vs code
* Drawing tools like i.e., draw.io

Each of the technologies are freely available and the technical skills required are manageable. Time limitations of the product development and the ease of implementing using these technologies are synchronized. Initially the web site will be hosted in a free web hosting space, but for later implementations it will be hosted in a paid web hosting space with a sufficient bandwidth. Bandwidth required in this application is very low, since it doesn’t incorporate any multimedia aspect

**Resource feasibility**

* Resources that are required for the project includes,
* Programming device (Laptop)
* Hosting space (freely available)
* Programming tools (freely available)
* Programming individuals(me)

**Financial Feasibility**

Being a web application movy will have an associated hosting cost. Since the system doesn’t consist of any multimedia data transfer, bandwidth required for the operation of this application is very low. The system will follow the freeware software standards. No cost will be charged from the potential customers. Bug fixes and maintaining tasks will have an associated cost. At the initial stage the potential market space will be around Nairobi and its enviros. Beside the associated cost, there will be many benefits for the clients and truck owners. Especially by reducing the cost of the truck services to an amount that is affordable to the common people and drivers can find clients quickly

1. **REQUIREMENT ELICITATION AND ANALYSIS**

**System review**

System review was used as technique to collect the requirements for the proposed system. This meant going through a number of systems similar to the proposed one, identifying how each of the system operated, what the systems had accomplished in production and what are the problems that these systems have not been able to address.

Below are the existing systems that were reviewed for this project:

1. Sendy Kenya
2. Cube Movers
3. Uber
4. DHL Movers

Sendy Kenya

Sendy is a delivery company in Kenya that provides a mobile app and web platform that enables individuals and small businesses to connect with Drivers and request on-demand or scheduled package delivery services anytime, any day, 24/7. The company offers bikes, tuk tuk, pick-ups, vans, and trucks as their delivery vehicles. It is available in Nairobi, Mombasa, Thika and Kisumu.

Cube Movers

Cube Movers is a smart and trusted removals and relocations company. It is currently Kenya’s leading premium removals service provider. It offers a variety of moving services; from pet relocation to office moving service. They have superior equipment and packaging to ensure your personal goods not only move in style but are also handled with utmost care. They have their own trucks for moving items.

Uber

Uber is the most popular taxi-hailing app right now. It is available in 630+ cities worldwide. It has achieved more than 100 million installs on playstore. With uber, you just tap to request a ride, and it’s easy to pay with credit or cash in select cities. Uber offers various types of cabs depending on the user’s preference. The most common is uberX for individual option. They have uberPOOL which you travel with other riders and reduce the cost. They also have UberBLACK for the high-end vehicles and more other options for their users.

DHL Movers

DHL Movers is a packers and movers company is a global company that has been at the front of delivering high quality services for diverse clients which include: Household Shifting, Commercial Shifting, Factory Shifting, Car carriers and Goods shifting.

Benefits of System Review:

1. It saves time since there is no need to go out to the field and collect data.
2. Data is readily available; on the internet, magazines, and journals, among others.
3. It is a cheaper way of collecting data as there are minimal costs involved.

**System development methodologies**

Agile Scrum Methodology

It is the most popular agile framework, an iterative approach that has at its core the sprint - the scrum term for iteration. In this development process sprints are set out. A sprint is a set period of time during which specific work has to be completed and made ready for review.

How it works:



Figure 3.1 - Agile Scrum model

1. During sprint planning, the team (student and supervisors) pulls a small chunk from the top of that wish list, a sprint backlog, and decides how to implement those pieces.
2. The team has a certain amount of time, a sprint, to complete its work – usually two to four weeks – but meets each day to assess its progress (daily scrum).
3. Along the way, the Scrum Master keeps the team focused on its goal.
4. At the end of the sprint, the work should be potentially shippable, meaning it can go to production.
5. The sprint ends with a sprint review and retrospective.
6. As the next sprint begins, the team chooses another chunk of the product backlog and begins working again.

Reasons as to why agile scrum was recommended for this project:

1. Fast moving, cutting edge developments can be quickly coded and tested using this method, as a mistake can be easily rectified. If a system is required to be developed within three-month agile scrum will enable coding of the system to be done within these three months.
2. There was clear visibility of the project development. This is because this method insists on frequent updating of the progress in work through regular meetings.
3. It became easier to cope with the changes due to short sprints and constant feedback
4. It was good for small, fast moving projects as it works well only with small team, which is perfect for this project managed by a single student with the help of supervisors.

Pitfall of agile scrum:

1. Agile Scrum is one of the leading scope creeps because unless there is a definite end date, the project management stakeholders will be tempted to keep demanding that new functionality be delivered.

**Data analysis tools**

For the data analysis, I used the following tools:

1. Graphs and Charts - Microsoft Excel was used to visually analyse the collected data

**System development tools**

For the proposed system, tools required were:

1. Machine: Laptop- 4 GB RAM and above
2. Operating System: Windows 7 and above /Linux/Unix
3. IDE: Visual Studio Code
4. Language: JavaScript
5. Stable internet connection.
6. Project Cost

Table 3.1 -Project Cost

|  |  |
| --- | --- |
| Resources | Cost (Kshs.) |
| Laptop | 50, 000 |
| Operating System | Free - 0 |
| Visual Studio Code | Free - 0 |
| Google maps api | Free trial |
| Database | Free trial - 0 |
| Internet | 5, 000 |
| TOTAL | 55, 000 |

1. **REQUIREMENT SPECIFICATION**

Hardware Requirements

Since it’s a web application it will be able to run on any device be it Desktop, Laptop or phone provided it connected to the internet. This will be able to create connection between the driver and the client

|  |  |  |
| --- | --- | --- |
| **SL** | **Hardware** | **Minimum System Requirement** |
| 01 | Processor | 2.4 GHz Processor speed |
| 02 | Memory | 2 GB RAM |
| 03 | Disk Space | 500 GB |

**Software Requirements**

The mobile phone should be of an android operating system, version 6 and above. The database is firebase. The phone should have location services

|  |  |  |
| --- | --- | --- |
| **Software** | **Minimum Requirement** | **Reason** |
| Operating System for mobile phone | Android 6 and above | Good support for android apps |
| Database Management System | Firebase | Easy to use and remote |
| Location services | Location | For maps functionality |

Functional requirements

* The application should allow for creating new accounts and logging in of users.
* The application should allow for editing and saving of user details
* The application should allow for making delivery requests by the clients
* The application should allow for receiving of the delivery requests from the drivers side
* The application should allow for cancelling of the delivery requests
* The application should allow for payment of the delivery by the client
* The application should allow for rating of the delivery by the client

**Non-functional requirements**

* User-friendly – The application user interface is simple and self-explanatory. All features are clear and the user does not have to search for them.
* Interactivity – The application prompts for input where needed and validates the format of the input. In case the input is incorrect, an error message is displayed.
* Performance – The application performs its operations such as locating where the user is without the causing the phone to hang or freeze.
* Adaptability – The application adapts to several android versions from version 6 to version 8 and works seamlessly.
* Efficiency – The application uses little memory for operation so that the phone does not crash

1. **REQUIREMENT VALIDATION**

The development of software begins once the requirements document is ‘ready’. One of the objectives of this document is to check whether the delivered software system is acceptable. For this, it is necessary to ensure that the requirements specification contains no errors and that it specifies the user’s requirements correctly. Also, errors present in the SRS will adversely affect the cost if they are detected later in the development process or when the software is delivered to the user. Hence, it is desirable to detect errors in the requirements before the design and development of the software begins. To check all the issues related to requirements, requirements validation is performed.

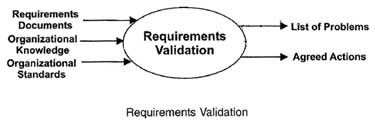
In the validation phase, the work products produced as a consequence of requirements engineering are examined for consistency, omissions, and ambiguity. The basic objective is to ensure that the SRS reflects the actual requirements accurately and clearly. Other objectives of the requirements document are listed below.

* To certify that the SRS contains an acceptable description of the system to be implemented
* To ensure that the actual requirements of the system are reflected in the SRS
* To check the requirements document for completeness, accuracy, consistency, requirement conflict ‘, conformance to standards and technical errors.

Requirements validation is similar to requirements analysis as both processes review the gathered requirements.Requirements validation studies the ‘final draft’ of the requirements document while requirements analysis studies the ‘raw requirements’ from the system stakeholders (users). Requirements validation and requirements analysis can be summarized as follows:

1. **Requirements validation:**Have we got the requirements, right?
2. **Requirements analysis:**Have we got the right requirements?

Various inputs such as requirementsdocument, organizational knowledge, and organizational standardsare shown. The requirements documentshould be formulated and organized according to the standards of the organization.The **organizational knowledge**is used to estimate the realism of the requirementsof the system. The **organizational standards are**specified standards followedby the organization according to which the system is to be developed.



## Requirements Review

Requirements validation determines whether the requirements are substantial to design the system. The problems encountered during requirements validation are listed below.

* Unclear stated requirements
* Conflicting requirements are not detected during requirements analysis
* Errors in the requirements elicitation and analysis
* Lack of conformance to quality standards.

To avoid the problems stated above, a **requirements review**is conducted, which consists of a review team that performs a systematic analysis of the requirements. The review team consists of software engineers, users, and other stakeholders who examine the specification to ensure that the problems associated with consistency, omissions, and errors are detected and corrected. In addition, the review team checks whether the work products produced during the requirements phase conform to the standards specified for the process, project, and the product.

At the review meeting, each participant goes over the requirements before the meeting starts and marks the items which are dubious or need clarification. Checklists are often used for identifying such items. Checklists ensure that no source of errors, whether major or minor, is overlooked by the reviewers. A ‘good’ checklist consists of the following.

* Is the initial state of the system defined?
* Is there a conflict between one requirement and the other?
* Are all requirements specified at the appropriate level of abstraction?
* Is the requirement necessary or does it represent an add-on feature that may not be essentially implemented?
* Is the requirement bounded and has a clear defined meaning?
* Is each requirement feasible in the technical environment where the product or system is to be used?
* Is testing possible once the requirement is implemented?
* Are requirements associated with performance, behaviour, and operational characteristics clearly stated?
* Are requirements patterns used to simplify the requirements model?
* Are the requirements consistent with the overall objective specified for the system/product?
* Have all hardware resources been defined?
* Is the provision for possible future modifications specified?
* Are functions included as desired by the user (and stakeholder)?
* Can the requirements be implemented in the available budget and technology?
* Are the resources of requirements or any system model (created) stated clearly?

The checklists ensure that the requirements reflect users’ needs and provide groundwork for design. Using the checklists, the participants specify the list of potential errors they have uncovered. Lastly, the requirements analyst either agrees to the presence of errors or states that no errors exist.