*Project Proposal*

**Group name:** Angry Pepe

**Group members:**

· Apoorv Joshi - 814003553

· Joshua O’Brien - 816001354

· Masood Khan - 05788124

**Project Name:** Kill it

**Project description:**

This software will allow users to have a controlled environment for an animal farm of any size and also a search and analysis database/ software of pests and diseases for farm animals locally and regionally.

The database would identify the type of disease the animal has, the different stages of the diseases as well as treatment options for the different stages of the diseases and where to find medication for said diseases.

The main objective of this project is to make a search and analysis software of pest and diseases of farm animals locally and regionally. The software made is going to be cross-platform software thus can run effectively in different environments enabling ease of use.

A person can search for various farm animal diseases with a search bar on the opened application. Once the user begins to input text into the search field they are presented with a drop list of search suggestions. This restriction allows the user to narrow and choose the content for which they are searching for. The software searches and analyzes the search and produces the outcome of the search.

The purpose of this software is aiding and arming the farmers with knowledge in identifying and mitigating the risk factors of pests and diseases of their livestock. This is extremely beneficial as they are able to make real time decisions based on the search outcome, thus reducing livestock losses due to spread of disease and securing their livelihood.

This software being made is a worthwhile, economically effective and reasonable endeavor. The data is easily stored in a cloud based environment therefore eliminating the need for buying and operating storage infrastructure. Information is readily available and accessible to users. Redundancy of data is greatly reduced. It also allows multiple users to access data. Data integrity is preserved as it ensures accuracy and consistency of data.

**Rationale for the project:**

In years gone farmers would observe their livestock having lesions or behaving strangely. By the time they are able to get help to figure out the cause, already their livestock would have taken a toll as they were unable to quickly identify the disease the animal was afflicted with. With the increase in technology, farmers would be able to search, analyze and identify the type of disease as well as the treatment options available utilizing our software, Kill It. This software is able to work across multiple device platforms as well as the software being user friendly.

**Business Benefits:**

* The software being made would be a product able to work across multiple device platforms. This enables ease of use for the user as the software is able to work efficiently in different environments without the user being constricted to one specific platform.

* The data is easily stored in a cloud based environment. This provides an economical advantage as it eliminates the need for buying, installing, operating and maintaining hardware storage infrastructure. Therefore server failures that cause downtime would no longer be an issue as information becomes readily available and accessible to users and also allows multiple users to access data. Data integrity is preserved as it ensures accuracy and consistency of data.
* Farmers utilize and benefit from the software technology with the software being able to analyze and identify the disease if their animals become afflicted. The software is able to identify and show treatment options available for the disease and its various stages. This results with the farmer being able to make real time decisions, containing any contagious diseases thereby mitigating the possibilities of outbreak and securing their livelihood.

**Stakeholders :**

User stakeholders: (Who will be using the system?)

The targeted users include but are not limited to

* Agricultural industry

Non-user stakeholders: (Who will not be directly using the system but benefits from the system?)

**User Environment: (Environment refers to the collection of hardware and software tools a system developer uses to build software systems.)**

* Android Operating System Firmware Version Oreo 8.1.0 (API 27) and upwards
* iOS 12 and upwards
* Android Studio

**Alternatives and Competition to your solution: (Are there ways the problem can be solved? Are there already products on the market that provide a solution? If yes what are they?)**

There is a product called

**Estimated Project Duration**

The project will be developed using a plan-driven strategy and is expected to be completed within 12 weeks.

**For each group member**

**Key skills (e.g. Databases, Web programming, etc.) –**

Apoorv Joshi - System Architecture Designer, MySQL database designer

Masood Khan - Mobile Application Programmer, Database Manager

Joshua O’Brien - Web programming, Design and analysis

**How each member contributes to the project**

* Joshua: Design a user friendly interface focused on maximizing usability and user experience
* Apoorv: Designing the system’s architecture to accommodate the best user experience.
* Masood: Implementing the backend and testing.
* All members will contribute towards the research and documentation.

**Statement of how you plan to work together (When you will meet, Collaboration tools you will use)**

Twice every week on Wednesdays and Saturdays at face to face conferences. The use of online collaboration tools such as google docs and whatsapp would be used to communicate on a regular day to day basis.

**User story 1**

The user opens the application and sees a search bar option available for searching. The user is able to type in " cattle disease " in the search bar where a drop list of search suggestions is presented to aid in spell checking and narrowing the field of search. Search type comes up with the name of the different types of diseases for different species of cattles locally and regionally along with images of the infected cattle. The searches returned in alphabetical order. User is then able to click on either the name of the disease or the pictures of the cattle. This action gives further details on the type and stages of the disease as well as treatment options available to treat the various stages of the disease or if cattle needs to be disposed off, if it's beyond medication as well as information on ethical ways of disposing of the cattle.

**User story 2**

The user is able to open the application on any smartphone with a built-in camera where there is an option available to take a picture of the animal. The user is able to take a picture and crop the image of the animal where the software will detect and aid in guiding the user in cropping the infected part of the animal . After cropping the image of the animal, the user is then able to send the image of the animal to the software database for analysis. The results of the analysis returns along with information identifying the type and stages of the disease as well as the veterinarian qualified to treat the disease. The user is able to tap on the veterinarian name thereby getting detailed information on the veterinarian name, address of veterinary clinic and contact information.

User story 3

The professional is able to request for a change in the database by sending a request email to the system admin. The user is able to

**Functional Requirements**

User Functional Requirements:

**Nb: specimen - plant,soil or other type of vegetation.**

1. The user shall be able to create an account.
2. The user shall be able to log into an existing account.
3. The user shall be able to upload a picture of the specimen to be analyzed.
4. The user shall be able to search the database for specific diseases, treatments and medications for a specific specimen.
5. The user shall be able to get further help from a professional when needed.

System Functional Requirements:

1. The system shall allow the user to create an account whereby providing an email address, contact number and a password.
2. The system shall allow the user to log into the system whereby providing their email address and password.
3. The system shall allow the user to update their password and contact information.
4. The system shall allow the user to take a picture using the device’s inbuilt camera and upload it to be analyzed.
5. The system will send a confirmation email to the user once a picture has been uploaded.
6. The system will email a detailed report to the user describing the type of disease, severity of the disease, possible treatments and where medication can be found.
7. The system will allow the user to query for information from the database which returns results based on popularity and alphabetical order.
8. The system shall be able to search the database by using the kill it software for data on a specific specimen.
9. The system shall allow the user to be able to upload a picture of a specimen to the kill it software which will then analyse the picture to give a response on whether or not there is information available in the database.
10. The system shall allow a user to be able to get additional help from a professional when a query to the database returns unsatisfactory results.
11. The system shall allow a user to be able to create a new account by entering a username using their email address and password.

**Non Functional Requirements**

1. The system shall email a detailed report to the user within 24 hours.
2. The software shall be able to allow users to login to the system using their username and password within a maximum time of two minutes.
3. The software shall be able to upload a maximum image size of 1 megabyte.
4. The software shall be able to complete analysis of the uploaded image by searching the database in at most three minutes.
5. The software shall be able to query the database for information in a maximum time of thirty seconds.

**USE CASES**

The **user** will be able to:

1. Register for the service
2. Upload a picture
3. Search the database

The **professional** will be able to:

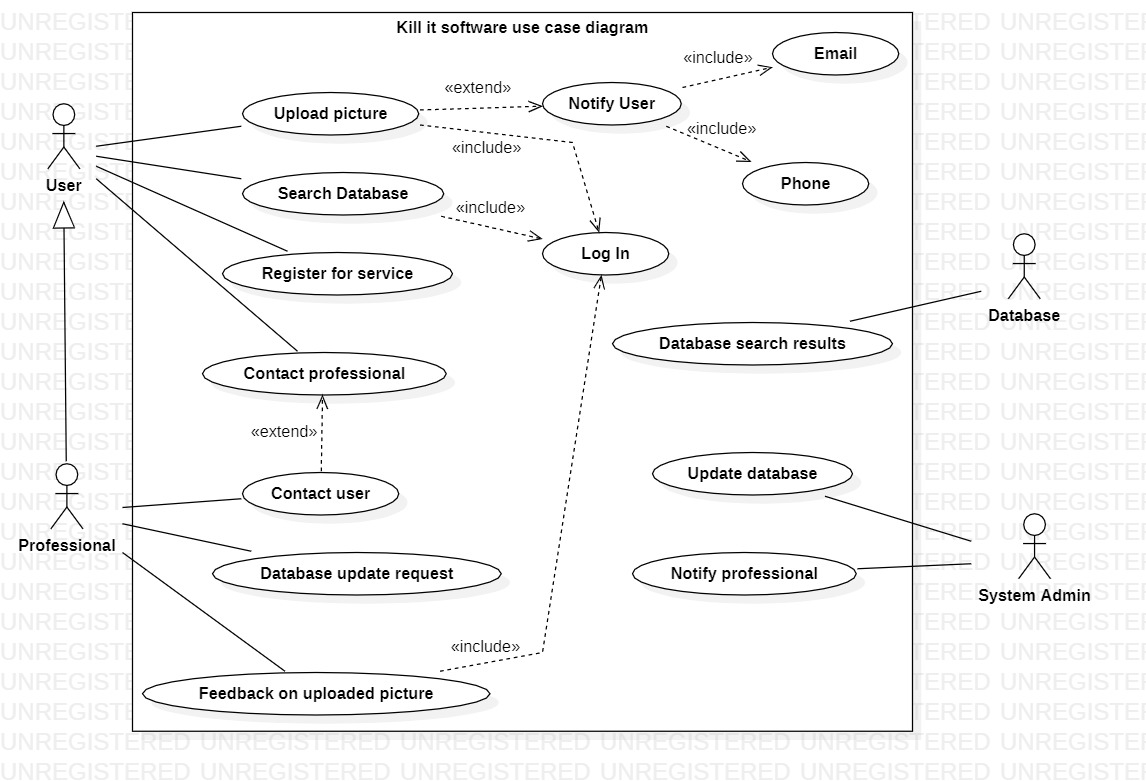
1. Register for the service
2. Query feedback
3. Contact the user for clarification
4. Request a database update if new information is found.

The **database** will be able to:

1. Provide detailed information (disease, treatment, medication, etc)

The **system admin** will be able to:

1. Update the database
2. Notify the professional for verification.



**Use case ranking and priority matrix**

The use case ranking and priority matrix is a tool used to evaluate use cases and determine their priority on a 1-5 scale against six criteria.

1. Significant impact on the architectural design.

2. Easy to implement but contains significant functionality.

3. Includes risky, time-critical, or complex functions.

4. Involves significant research; new or risky technology.

5. Includes primary business functions.

6. Will increase revenue or decrease costs.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Use case ID | Use case name | Ranking criteria, 1-5 | | | | | | Total score | Priority | Build cycle |
|  |  | 1 | 2 | 3 | 4 | 5 | 6 |  | | |
| 1 | Register for service | 5 | 5 | 1 | 1 | 3 | 3 | 18 | Medium | 1 |
| 2 | login | 5 | 5 | 1 | 1 | 3 | 3 | 18 | Medium | 2 |
| 3 | Search database | 4 | 4 | 4 | 1 | 5 | 5 | 23 | High | 3 |
| 4 | Upload picture | 4 | 5 | 3 | 1 | 5 | 5 | 23 | High | 4 |
| 5 | Feedback on uploaded picture | 3 | 5 | 2 | 1 | 5 | 2 | 18 | High | 5 |
| 6 | Database search results | 5 | 1 | 4 | 2 | 3 | 1 | 16 | High | 5 |
| 7 | Contact User | 3 | 3 | 4 | 1 | 3 | 1 | 15 | Medium | 6 |
| 8 | Contact Professional | 3 | 2 | 3 | 1 | 1 | 1 | 11 | Low | 6 |
| 9 | Request database update | 2 | 5 | 2 | 1 | 1 | 2 | 13 | Medium | 7 |
| 10 | Update database | 2 | 2 | 2 | 1 | 2 | 1 | 10 | High | 8 |
| 11 | Notify professional | 1 | 1 | 1 | 1 | 1 | 1 | 6 | Low | 9 |

**EXPANDED USE CASE AND SEQUENCE DIAGRAM**

Expanded Use Case

Kill It System

|  |  |
| --- | --- |
| Author(s): Apoorv Joshi  Joshua O’Brien  Masood Khan | Date: 18th March 2019  Version: 1.0 |

|  |  |  |
| --- | --- | --- |
| **Use-Case Name** | Upload Picture | Use-Case Type: Business |
| **Use-Case-ID** | 3 |
| **Priority** | High |
| **Source** |  |
| **Primary Business Actor** | User | |
| **Other Participating Actors** | Professional | |
| **Other Interested Stakeholders** | Professional - Interested in information authenticity. | |
| **Description** | This use case describes the event of a user uploading a picture of the pest or disease affecting their farm. Upon uploading the picture, the notification system notifies the user via email or text message. The uploaded picture is then in a queue waiting to be analysed by a professional. | |
| **Precondition** | The user must be registered for the service. | |
| **Trigger** | Initiated when the user uploads a picture. | |
| **Typical Course of Events** | **Actor Action** | **System Response** |
| Step1: The user uploads a picture to the database. | Step 2: The database sends a confirmation to the Notification System.    Step 3: The Notification System sends a notification to the user via email or text message, depending on the user’s preference. |
| **Alternative Courses** |  | |
| **Conclusions** | Concludes when the system notifies the user. | |
| **Post-condition** | The Professional, based on their opinion, sends the user a feedback report on the picture uploaded. | |
| **Business Rules** | Availability of professionals may affect the amount of time taken to get feedback to the user. | |
| **Implementation Constraints and Specifications** | The size of the image uploaded by the user must not exceed the specified size. | |
| **Assumptions** |  | |
| **Open Issues** |  | |

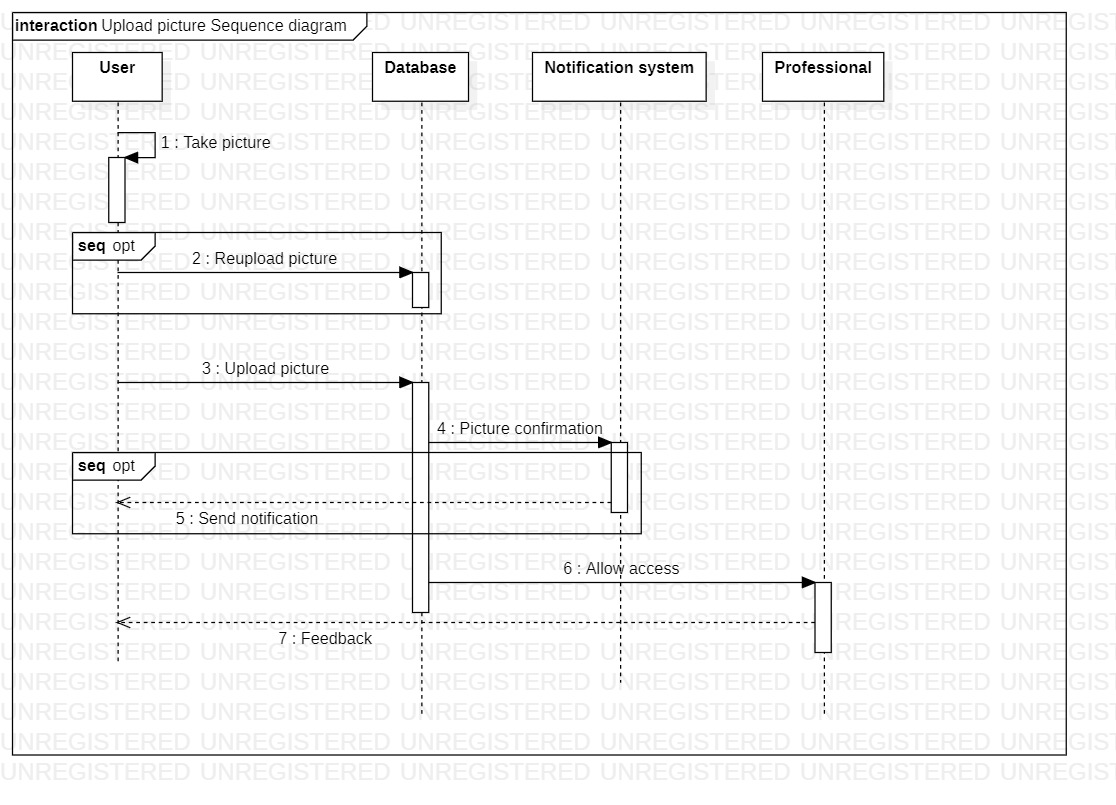
|  |  |
| --- | --- |
| Author(s): Apoorv Joshi  Joshua O’Brien  Masood Khan | Date: 18th March 2019  Version: 1.0 |

|  |  |  |
| --- | --- | --- |
| **Use-Case Name** | Register for Service | Use-Case Type: Business |
| **Use-Case-ID** | 1 |
| **Priority** | Medium |
| **Source** |  |
| **Primary Business Actor** | User | |
| **Other Participating Actors** | Professional | |
| **Other Interested Stakeholders** | N/A | |
| **Description** | This use case describes the event of a person registering for the service whether it be a user or a professional. Upon successfully registering by providing a username and a password, the system saves the account information. | |
| **Precondition** | none. | |
| **Trigger** | Initialized when a person clicks the register button. | |
| **Typical Course of Events** | **Actor Action** | **System Response** |
| Step1: the user clicks the register button. | Step 2: the system adds the account to the accounts database. |
| **Alternative Courses** | none. | |
| **Conclusions** | Concludes when the system notifies the user that they have successfully registered for the service. | |
| **Post-condition** | the logon screen will be displayed again for the person to log into the system using their newly created credentials. | |
| **Business Rules** | Username may not be available due to being already taken or used in another account. | |
| **Implementation Constraints and Specifications** | Passwords must have special characters and integers and be a maximum of 8 characters. | |
| **Assumptions** |  | |
| **Open Issues** |  | |

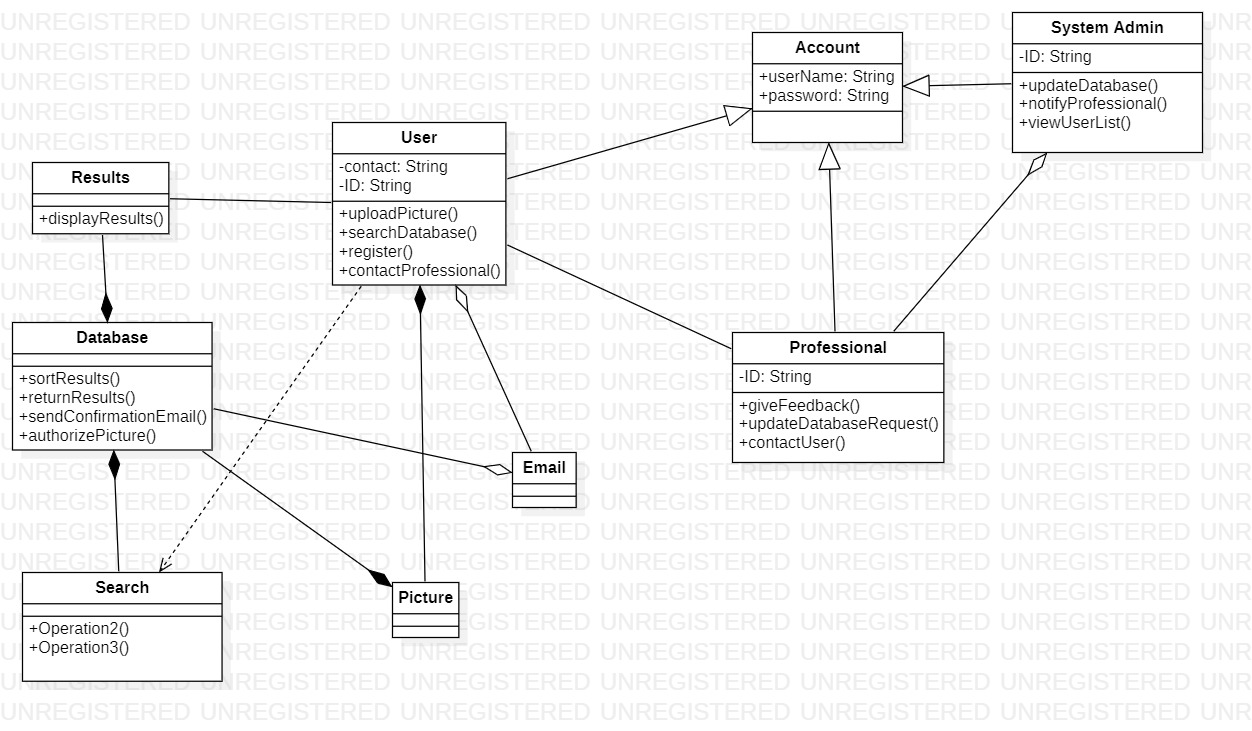
|  |  |
| --- | --- |
| Author(s): Apoorv Joshi  Joshua O’Brien  Masood Khan | Date: 18th March 2019  Version: 1.0 |

|  |  |  |
| --- | --- | --- |
| **Use-Case Name** | Request database update | Use-Case Type: Business |
| **Use-Case-ID** | 10 |
| **Priority** | Medium |
| **Source** |  |
| **Primary Business Actor** | Professional | |
| **Other Participating Actors** | System Admin | |
| **Other Interested Stakeholders** | Database | |
| **Description** | This use case describes the event where a professional sends an “update database” request to the system admin in the case where new information is found and needs to be added to the database in order to keep it up to date. | |
| **Precondition** | none. | |
| **Trigger** | Initialized when the professional contacts the system admin. | |
| **Typical Course of Events** | **Actor Action** | **System Response** |
| Step 1: The professional clicks on the “send update request” button.  Step 3: The system admin emails the professional to get the details in order to update the database.  Step 4: The professional provides the details to the System Admin  Step 5: The system admin updates the database with credit to the professional | Step 2: The system notifies the system admin of the new update request from the professional.    Step 6: The system sends the professional a notification of the updated database entry suggested by him. |
| **Alternative Courses** | none. | |
| **Conclusions** | Concludes when the system notifies the professional of the database update. | |
| **Post-condition** | The information added to the database after the update will now be visible to users in their searches. | |
| **Business Rules** |  | |
| **Implementation Constraints and Specifications** | The promptness of the system admin will be crucial to how quickly the database can be updated. | |
| **Assumptions** |  | |
| **Open Issues** |  | |

**Sequence Diagram** for the use case with the highest score.

****

**CLASS DIAGRAM**

****

**TESTING**

Method: searchDatabase(String disease, String TreatmentType, String medication, Array database)

|  |  |  |  |
| --- | --- | --- | --- |
| Criterion | Valid Input | Invalid Input | Boundary |
| Disease | A string containing the name of the disease | Anything other than the name of the disease. |  |
| Treatment Type | A string containing the name of the treatment | Anything other than the name of the treatment |  |
| Medication | A string containing the name of the medication | Anything other than the name of the medication |  |
| database | An array of dictionaries | Anything other than the required input |  |

Method: contactUser(String contact)

|  |  |  |  |
| --- | --- | --- | --- |
| Criterion | Valid Input | Invalid Input | Boundary |
| Contact | A string containing the contact of the user. | Any string or data type other than required. |  |

Method: updateDatabase(String password, Array database)

|  |  |  |  |
| --- | --- | --- | --- |
| Criterion | Valid Input | Invalid Input | Boundary |
| password | A string containing the password of the system admin | Any string or data type other than required. |  |
| database | An array of Dictionaries | Any array or data that is not of Dictionary type |  |

Method: viewUserUserList(Array userList)

|  |  |  |  |
| --- | --- | --- | --- |
| Criterion | Valid Input | Invalid Input | Boundary |
| userList | An array of strings containing the id of the user. | Any data type other than an array of strings. |  |

Method: sendConfirmationEmail(String userEmailID)

|  |  |  |  |
| --- | --- | --- | --- |
| Criterion | Valid Input | Invalid Input | Boundary |
| userEmailID | A string with the user email ID | Anything else |  |

**RISK MANAGEMENT**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Risk (Project, Business, Product)** | **Affects** | **Effects** | **Probability** | **Strategy Category** | **Strategies to be employed** |
| **Product Competition:** Another company develops a similar website/application | Business | Serious | Moderate | Minimization | Employ better marketing strategies to encourage user participation or buy out the competition. |
| **Staff Turnover:** Experienced staff members leaving the project before it is fully developed. | Project | Serious | Very Low | Avoidance | Keeping additional staff members on standby in the event a staff member leaves. |
| **Developmental Delays:**  The amount of time required to complete the project is underestimated | Project | Tolerable | Moderate | Contingency and Avoidance | Proper management is essential. |
| **Rapid Customer Growth:**  A larger than expected increase in user traffic may cause a server overload | Product | Serious | High | Contingency | Using self scaling servers in order to accommodate the increase in user traffic |
| **Database or Server Failure:**  Failure results in the software not working as intended | Project | Catastrophic | Very Low | Contingency | Backup servers can be used. |
| **Estimation:**  The final product’s size is underestimated. | Product and Project | Tolerable | Moderate | Avoidance | Efficient programming |
| **Qualified Professionals:**  Feedback should be given to the users from qualified professionals only. | Product and Business | Serious | Moderate | Minimization | Requesting proof of qualification from said professionals at the time of registering |
| **Management Change:**  A change occurring in management will result in differing management priorities | Project | Management | Low | Contingency | A priority chart made at the beginning of development will help guide all managers |
| **CASE tools underperformance:**  Diagrams cannot be integrated or linked due to compatibility | Product | Tolerable | High | Avoidance | Use or acquire tools from the same software supplier that has a full line of products to ensure future compatibility if more tools are added |
| **Gold Plating:**  Developers adding functionality and design not required for the software | Project | Tolerable | Low | Avoidance | Proper management will help keep the development team on track. |
| **Time loss during development:**  Developers losing time on installing dependencies | Project | Tolerable | Low | Avoidance | The software can be developed through a container platform (eg: docker) so that developers don’t have to waste time installing dependencies. |
| **Strategic Risk:**  The software no longer applies to farm related diseases. | Product | Serious | Low | Avoidance | The software can be evolved to become a |
| **Financial Mismanagement:**  Exhausting funds earlier than predicted, resulting in failure of further development | Project and Product | Serious | Low | Avoidance | Provide senior management with a documented report of the costs and keep them updated on expenses incurred |
| **Component Integration:**  Components of the software stop working | Product and Business | Catastrophic | Low | Avoidance | Detailed inspection and testing of the software should be conducted before deployment to ensure all components work effectively |
| **SQL Injection:**  Hackers gain access to the database and can perform malicious attacks | Business | Catastrophic | Low | Contingency | A secure biometric system (example: fingerprint scanning) along with password authentication |

## 

## **Cost Estimation**

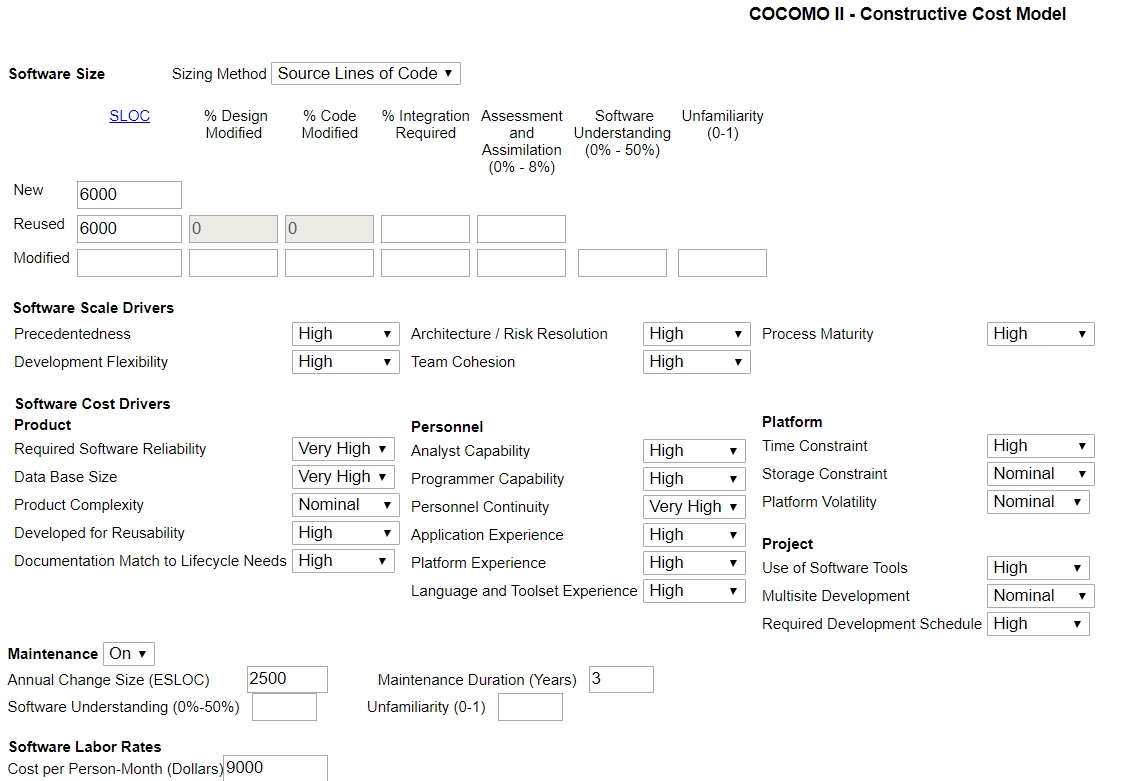
**Specification and justification for cost modelling technique**

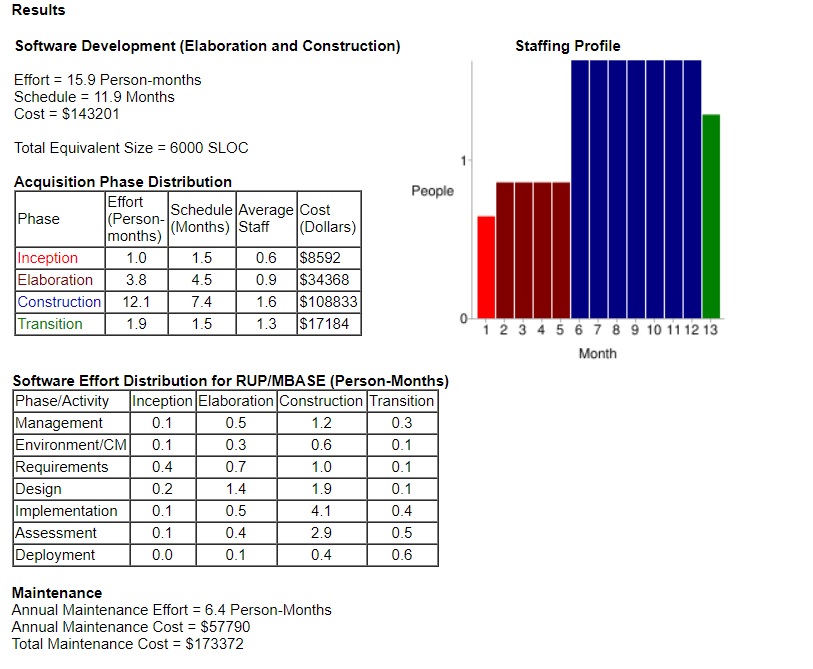
The COCOMO II (Constructive Cost Model II) is a software cost estimation model used to estimate the cost, effort and schedule for a new software development activity. The model consists of three sub models that offer increasingly detailed accuracy. These include Application Composition used for early prototyping efforts and Early Design and Post-Architectural models used further into the development life cycle. The COCOMO II model derives its estimations based on estimated project size, a number of scale drivers and software cost drivers that assess the project, development environment, and project team.

The COCOMO II model is very well suited to our cost estimation requirements. COCOMO II’s use of industry data was very helpful in terms of getting an approximate starting figure. The generated estimations will allow the project management team to have a very clear and in depth understanding of the budget breakdown, how to determine a development team size and meet project deadlines.

The project cost estimation will also give the team an idea of how to approach investors along with determining how many investors will be required to finance the project. The estimated costs can be weighed against the estimated profits in efforts to further persuade investors with regard to the feasibility and profitability of the project.

**Project cost (breakdown)**





**REFERENCES**

http://www.utm.mx/~caff/doc/OpenUPWeb/openup/guidances/guidelines/writing\_good\_requirements\_48248536.html

Sommerville, Ian (2016). Software Engineering (10th ed.). Pearson Education. *ISBN*

*978-0-13-394303-0.*

http://softwarecost.org/tools/COCOMO/

http://softwarecost.org/tools/COCOMO/data/COCOMO\_April\_13\_2020\_02\_40\_33\_389775.

https://myelearning.sta.uwi.edu/pluginfile.php/874319/mod\_resource/content/0/Cost%20Estimation.pdf