Jarvis Emulator  
Software Requirements Specification  
COP 4331, Fall 2015

**Modification History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Who** | **Comment** |
| v0.0 | 10/3/2015 | Robin Schiro | Created document |
| v1.0 | 10/4/2015 | Jimmy Lam | Added more to introduction and product overview, my requirements, and my events to the event table |
| v1.1 | 10/4/2015 | Robin Schiro | Modified functional reqs so that all tables use a numbered list |
| v1.3 | 10/5/2015 | Manuel Gonzalez | Added functional requirements specific to speech construction, and sections 3.7, 3.8 and 3.9 of requirements and events specific to speech construction |

**Team Members:**

* Jimmy Lam
* Julian Rojas
* Manuel Gonzalez
* Robin Schiro

1. **Introduction**
   1. **Software to be Produced**
      1. The goal of this project is to produce a Windows desktop application called the Jarvis Emulator to enhance the desktop experience. This application can detect and recognize users as they enter or exit the room, and can respond to the users’ voice commands though speech construction and perform tasks for the user, such as opening other applications, taking a picture of the users, displaying relevant information to the users through website APIs, and logging the users in and out of their computer. The front end GUI will allow the users to set up their profiles, train Jarvis to remember their faces, and configure other settings of the application. It will also display visuals to let the users know it is speaking to the users.
   2. **Reference Documents**
      1. Please refer to Project Management Plan and Concept of Operations
   3. **Applicable Standards**
      1. **Testing Standard**
         1. We will create our test cases to be reasonable enough so that we can fairly evaluate the performance of our application
         2. We will log our tests to keep track of our progress and make sure that our project is fully functional
      2. **Coding Standard**
         1. We will write our codes in classes so that we can each test our sections of the project and easily integrate our work into one cohesive project later on
   4. **Definitions, Acronyms, and Abbreviations**
      1. **Definitions**
         1. Trained user: For all test cases, a “trained user” is one who has provided sufficient training data for his/her profile. This means that at least 50 pictures of his/her face at various angles have been captured by the application.
         2. Active user: The user who currently has control over the application.
      2. **Acronyms**
         1. API: Application Programming Interface
2. **Product Overview**
   1. **Assumptions**
      1. As the developers, we assume that Jarvis Emulator users will be using a Windows Desktop computer. We assume that users will have internet connection for the website APIs used by Jarvis. We also assume that users will have a webcam with a high enough resolution for Jarvis to recognize their face and a microphone with high enough quality to detect the user’s voice correctly.
   2. **Stakeholders**
      1. **Clients**
         1. Dr. Turgut, our professor, expects that we finish a presentable project and to be impressed by our work. She also has high hopes that all of her students do well in the course
         2. Amirreza Samiei, our grader, overlooks our project and will make sure we meet our requirements for the project. He also hopes that we are able to do well in the class
      2. **Developers**
         1. We, the developers are also stakeholders of this project as our grade in the class depends on it. Since we also came up with the project idea, we want to make sure our project works well and impresses the users of this application.
   3. **Event Table**

|  |  |  |  |
| --- | --- | --- | --- |
| *Event Name* | *External Stimuli* | *External Responses* | *Internal data and state* |
| User Recognition | Trained user enters the view of the webcam | The application will greet the user and inquire about what the user would like it to do. | The application will store the name of the active user and wait for commands from this user. |
| User Inputting Configuration Settings | The user opens the ‘Configuration’ tab and input information into the settings fields. | The application saves inputted settings once the user hits ‘Save’. | The application will store the settings in the user’s profile file. While the user is inputting information, the application is still listening for commands. |
| Login | User enters room | Logs into computer and shows the starting page | Recognizes user face and account by going through its trained faces file |
| Logout | User voice command | Logs out of computer into login screen | Logs user out and wait for user to return to log in |
| Open other applications | User voice command | Open desired applications | Searches for the application and open for user |
| Taking picture | User voice command | Takes picture of user | Uses the webcam to take picture of user and stores the picture |
| Data description through Natural Language | User voice commands to output specific data | Speaks to the user with the information requested | Specifically formatted data is fed into the speech construction module for Natural Language Generation. Upon completion, the generated text is output in audio form using Text-To-Speech algorithms. |
| Response to user question | User voice command parsed as a question | Speaks to the user with the answer to the question or that it doesn’t know the answer | The speech recognition module will parse the command, detect that is a question, check whether or not is supported by the system and post to all modules the desired response, which ultimately will lead to feeding information to the speech construction module. |
| Notification of System Event | An important event happens within the system | Speaks to the user about the event that just happened | Once an event is detected, the speech construction module will generate and output a notification accordingly. |

* 1. **Use Case Diagram**
  2. **Use Case Descriptions**

1. **Specific Requirements**
   1. **Functional Requirements**

|  |  |
| --- | --- |
|  | |
| Statement: | The frames of the feed are processed under eigenanalysis using the OpenCV library. Recognition occurs with a minimum of 70% accuracy. |
| Source: | Developers |
| Dependency: | None |
| Conflicts: | None |
| Supporting Materials: | [Source of algorithm](http://www.codeproject.com/Articles/239849/Multiple-face-detection-and-recognition-in-real) |
| Evaluation Method: | Set up the application with at most five trained users, with you being one of them. Then, exit and enter the view of the webcam 10 times. The application should recognize you at least 7 of the 10 times. |
| Revision History: | Robin Schiro | 10/3/15 | Created the requirement |

|  |  |
| --- | --- |
|  | |
| Statement: | The application can track the position of the user’s face. |
| Source: | Developers |
| Dependency: | None |
| Conflicts: | None |
| Supporting Materials: | [Source of algorithm](http://www.codeproject.com/Articles/239849/Multiple-face-detection-and-recognition-in-real) |
| Evaluation Method: | Test Cases 2 |
| Revision History: | Robin Schiro | 10/3/15 | Created the requirement |

|  |  |
| --- | --- |
|  | |
| Statement: | The user interface allows the user to “train” the application for facial recognition. |
| Source: | Developers |
| Dependency: | Requirement 2 |
| Conflicts: | None |
| Supporting Materials: | None |
| Evaluation Method: | Test Cases 3 and 4 |
| Revision History: | Robin Schiro | 10/3/15 | Created the requirement |

|  |  |
| --- | --- |
|  | |
| Statement: | The user interface allows the user to save and update set of configuration settings based on selections made in the ‘Configuration’ tab. |
| Source: | Developers |
| Dependency: | None |
| Conflicts: | None |
| Supporting Materials: | None |
| Evaluation Method: | Test Cases 5 |
| Revision History: | Robin Schiro | 10/3/15 | Created the requirement |

|  |  |
| --- | --- |
|  | |
| Statement: | Jarvis shall recognize voice commands of the user using window’s speech library and should have 80% accuracy |
| Source: | Developers |
| Dependency: | None |
| Conflicts: | None |
| Supporting Materials: | This YouTube video: <https://www.youtube.com/watch?v=KR0-UYUGYgA> |
| Evaluation Method: | Test Case 6 |
| Revision History: | Jimmy Lam | 10/4/15 | Created the requirement |

|  |  |
| --- | --- |
|  | |
| Statement: | Jarvis shall open other applications based on user command |
| Source: | Developers |
| Dependency: | No 1 |
| Conflicts: | None |
| Supporting Materials: | None |
| Evaluation Method: | Test Case 7 |
| Revision History: | Jimmy Lam | 10/4/15 | Created the requirement |

|  |  |
| --- | --- |
|  | |
| Statement: | Statement: Jarvis shall log the user in and out of their computer based on user command |
| Source: | Developers |
| Dependency: | No 1 |
| Conflicts: | None |
| Supporting Materials: | None |
| Evaluation Method: | Test Cases 8 and 9 |
| Revision History: | Jimmy Lam | 10/4/15 | Created the requirement |

|  |  |
| --- | --- |
|  | |
| Statement: | Statement: Jarvis shall take a picture of the user for the user and store the photo |
| Source: | Developers |
| Dependency: | No 1 |
| Conflicts: | None |
| Supporting Materials: | None |
| Evaluation Method: | Test Case 10 |
| Revision History: | Jimmy Lam | 10/4/15 | Created the requirement |

|  |  |
| --- | --- |
|  | |
| Statement: | Given specifically formatted data, the application should generate human language speech that summarizes and describes the data |
| Source: | Developers |
| Dependency: | None |
| Conflicts: | None |
| Supporting Materials: | None |
| Evaluation Method: | Test Case 12 |
| Revision History: | Manuel Gonzalez | 10/5/15 | Created the requirement |

|  |  |
| --- | --- |
|  | |
| Statement: | The application should answer basic user questions |
| Source: | Developers |
| Dependency: | No. 5 and 9 |
| Conflicts: | None |
| Supporting Materials: | None |
| Evaluation Method: | Test Case 13 |
| Revision History: | Manuel Gonzalez | 10/5/15 | Created the requirement |

|  |  |
| --- | --- |
|  | |
| Statement: | The application should greet the user through the speakers upon user recognition |
| Source: | Developers |
| Dependency: | No 1 |
| Conflicts: | None |
| Supporting Materials: | None |
| Evaluation Method: | Test Case 14 |
| Revision History: | Manuel Gonzalez | 10/5/15 | Created the requirement |

* 1. **Resource Requirements**

|  |  |
| --- | --- |
| *No: 12* | |
| Statement: | The application needs a camera and microphone to function |
| Source: | Developers |
| Dependency: | None |
| Conflicts: | None |
| Supporting Materials: | None |
| Evaluation Method: | None |
| Revision History: | Manuel Gonzalez | 10/5/15 | Created the requirement |

* 1. **Security Requirements**

|  |  |
| --- | --- |
| *No: 13* | |
| Statement: | The application should be able to correctly identify users with a 70% accuracy and separate user specific information |
| Source: | Developers |
| Dependency: | No. 1 |
| Conflicts: | In case is not possible to identify users with 70% accuracy, the user will be prompted to provide his/her name |
| Supporting Materials: | None |
| Evaluation Method: | None |
| Revision History: | Manuel Gonzalez | 10/5/15 | Created the requirement |

|  |  |
| --- | --- |
| *No: 14* | |
| Statement: | All user information will be backed up in the Windows OS’s Appdata to prevent loss of information when updating or reinstalling the application |
| Source: | Developers |
| Dependency: | None |
| Conflicts: | None |
| Supporting Materials: | <http://windows.microsoft.com/en-us/windows-8/what-appdata-folder> |
| Evaluation Method: | None |
| Revision History: | Manuel Gonzalez | 10/5/15 | Created the requirement |

* 1. **Quality Assurance Requirements**

|  |  |
| --- | --- |
| *No: 15* | |
| Statement: | In case of a system fault, the system should notify the user and attempt to reset after 1 min of non-response |
| Source: | Developers |
| Dependency: | None |
| Conflicts: | None |
| Supporting Materials: | None |
| Evaluation Method: | Test Case 11 |
| Revision History: | Manuel Gonzalez | 10/5/15 | Created the requirement |

|  |  |
| --- | --- |
| *No: 16* | |
| Statement: | The system should have a mean time between faults of 7 days, roughly failing once every week of continuous use |
| Source: | Developers |
| Dependency: | None |
| Conflicts: | None |
| Supporting Materials: | None |
| Evaluation Method: | None |
| Revision History: | Manuel Gonzalez | 10/5/15 | Created the requirement |

|  |  |
| --- | --- |
| *No: 17* | |
| Statement: | The system shouldn’t allocate more than 1 GB of memory usage |
| Source: | Developers |
| Dependency: | None |
| Conflicts: | None |
| Supporting Materials: | None |
| Evaluation Method: | Test Case 16 |
| Revision History: | Manuel Gonzalez | 10/5/15 | Created the requirement |

|  |  |
| --- | --- |
| *No: 18* | |
| Statement: | The system should have an average response time no longer than 5 seconds. If longer than that the user should be notified that the current activity may take longer than usual |
| Source: | Developers |
| Dependency: | None |
| Conflicts: | None |
| Supporting Materials: | None |
| Evaluation Method: | Test Case 15 |
| Revision History: | Manuel Gonzalez | 10/5/15 | Created the requirement |