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 |
| v1.4 | 10/5/2015 | Julian Rojas | Added the last functional requirements & section 3.3, 3.4, 3.5, 3.6. Also added use case descriptions and my events |
| v1.5 | 10/6/2015 | Jimmy Lam | Added user to stakeholders |
| v1.6 | 10/6/2015 | Robin Schiro | Inserted new Use Case Diagram, updated ref doc links |

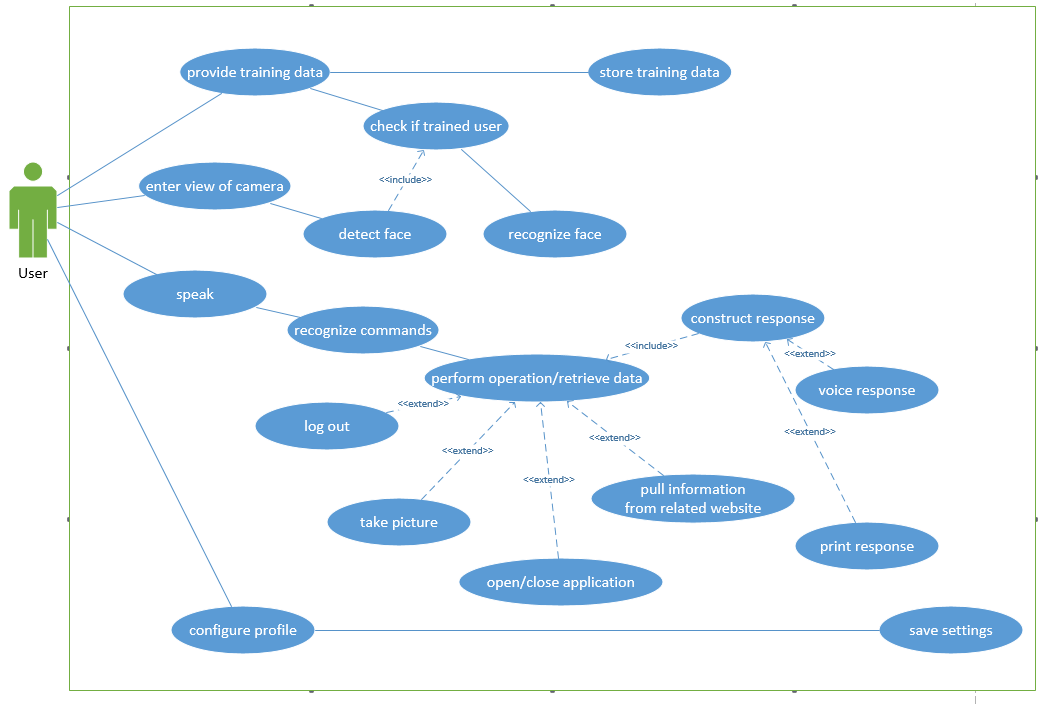
**Team Members:**

* 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. It can also perform tasks for users, such as opening other applications, taking a picture of the users, displaying relevant information to the users through website RSS feeds, and logging users 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. [Project Management Plan](https://www.dropbox.com/s/z3p8gxvhr7t3vlk/Project%20Management%20Plan.docx?dl=0)
      2. [Concept of Operations](https://www.dropbox.com/s/uhnp6yg3z2rm68f/Concept%20of%20Operations.docx?dl=0)
      3. [Test Plan](https://www.dropbox.com/s/1fira47foaf9itv/Test%20Plan.docx?dl=0)
   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 code 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. We assume that a Jarvis Emulator user will be using a Windows computer. We assume that the user will have an internet connection for the RSS feeds used by Jarvis. We also assume that the user will have a webcam with a high enough resolution for Jarvis to recognize his face and a microphone with high enough quality to detect his 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.
         3. Normal Users: The users of the application expect our application to be easy to use and expect that it will improve their desktop experience with easy voice commands and relevant information feeds.
      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 inputs 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. |
| 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. |
| Weather forecast | User voice commands | Application displays/speaks to user | Jarvis will access weather RSS Feed/ API and it will pull information regarding the forecast |
| News Headlines | User Voice Commands | Application displays/speaks to user | Jarvis will access news website’s RSS feed and pull articles headlines. It will give a link to full article |
| Mouth movement | User’s mouth movement within box outline | Detects active user | Application detects who is talking if multiple users are present |

* 1. **Use Case Diagram**



* 1. **Use Case Descriptions**
  + **Enter View Of Camera**
    - User is within 5 feet of webcam.
    - **Exception:** User is not facing camera
  + **Recognize Face**
    - Application tries to recognize user with 70% accuracy
    - **Exception:** User has not been trained or lighting conditions are not adequate.
  + **Recognize Commands**
    - Jarvis is able to recognize keywords
    - **Exception:** Commands is not already specified.
  + **Configure Profile**
    - User is able set up profile, add preferences, and applications
    - **Exception:** User has not been recognized
  + **User Speaks**
    - The user will communicate with Jarvis using voice commands
    - **Exception:** Jarvis is not able to recognize user’s speech
  + **Provide Training Data:**
    - User takes different pictures and provide them to the application through user interface
    - **Exception:** User has not been recognized
  + **Store Training Data**
    - Jarvis will save user pictures
    - **Exception**: The user is not in view of the webcam, there is no space available.
  + **Check If Trained User**
    - Jarvis will check if a detect face is a trained user or a new user
    - **Exception:** Jarvis is not able to detect face
  + **Detect Face**
    - Jarvis will analyze frame if face has entered the scene.
    - **Exception:** Environmental conditions prevent detection from occurring (i.e. poor lighting)
  + **Perform Operation**
    - Jarvis will take commands as input and complete the related task
    - **Exception:** The command is not supported by the application.
* **Log out**
* Jarvis will log the user out of his or her computer
* **Exception**: The user is not logged in
* **Take** **picture**
* Jarvis will take a photo of the user and store the picture in a file
* **Exception**: The user is not in view of the webcam, there is no space available.
* **Open/close application**
* Jarvis will open and close applications based on user command
* **Exception**: The desired application is not supported or is not defined in a path for Jarvis
* **Pull information from related websites**
* Jarvis will use RSS to display to user the information from websites
* **Exception**: The website is not available or does not support RSS
* **Voice** **response**
* Jarvis will perform text to speech of its response construction
* **Exception**: Jarvis does not construct a response
* **Print Response**
* Jarvis will display its response to the GUI
* **Exception**: Jarvis did not construct a response
* **Save profile**
* Jarvis will save the user’s profile to a text file
* **Exception**: The user did not select to save his or her profile
* **Construct Response**
* Jarvis will construct a response from a grammar to display to user
* **Exception**: Jarvis does not recognize user voice

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 70% 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: | Jarvis shall log the user out of their computer based on user command. |
| Source: | Developers |
| Dependency: | No 1 |
| Conflicts: | None |
| Supporting Materials: | None |
| Evaluation Method: | Test Case 8 |
| Revision History: | Jimmy Lam | 10/4/15 | Created the requirement |

|  |  |
| --- | --- |
|  | |
| 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 11 |
| 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 12 |
| 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 1 |
| Revision History: | Manuel Gonzalez | 10/5/15 | Created the requirement |

|  |  |
| --- | --- |
|  | |
| Statement: | Jarvis should be able to subscribe to Website RSS Feeds. |
| Source: | Developer |
| Dependency: | No. 5 |
| Conflicts: | None |
| Supporting Materials: | None |
| Evaluation Method: | Test Cases 15 & 16 |
| Revision History: | Julian Rojas | 10/5/15 | Created the requirement |

|  |  |
| --- | --- |
|  | |
| Statement: | Jarvis should be able to detect mouth movement. |
| Source: | Developer |
| Dependency: | No. 1 & No. 2 |
| Conflicts: | None |
| Supporting Materials: | None |
| Evaluation Method: | Test Case 17 |
| Revision History: | Julian Rojas | 10/5/15 | Created the requirement |

**3.2 Interface Requirements**

|  |  |
| --- | --- |
|  | |
| Module: | Speech Construction |
| Input: | Events, Formatted data from websites (RSS feeds), |
| Output: | Output: Text and audio of natural-language sentences |
| Data Type: | Input: String  Output: MP3 |
| Accuracy: | N/A |
| Receive/Send Frequency: | Depends on how often the user speaks |
| Timing Issues: | Asynchronous; Post status on regular basis (i.e. each second) |
|  | |
| Module: | Speech Recognition |
| Input: | User voice commands |
| Output: | The operations that need to be performed |
| Data Type: | Output: Result of action (e.g. opening application, logging out, etc.) |
| Accuracy: | Output: 75% accuracy on parsing commands |
| Receive/Send Frequency: | Depends on how often the user speaks |
| Timing Issues: | Asynchronous; Post status on regular basis (i.e. each second) |

|  |  |
| --- | --- |
|  | |
| Module: | Web Access |
| Input: | The URL of the website RSS feed |
| Output: | Formatted data from websites (RSS feeds) that the speech construction module can process |
| Data Type: | Input: URL (string)  Output: XML document |
| Accuracy: | Output: Will depend on the accuracy of the RSS feed |
| Receive/Send Frequency: | One web access request and one response each time the user requests an update from a website |
| Timing Issues: | Asynchronous; Post status on regular basis (i.e. each second) |

|  |  |
| --- | --- |
|  | |
| Module: | Facial Recognition |
| Input: | Video feed from webcam |
| Output: | A list of recognized users and the positions of their faces in the frame |
| Data Type: | Output: List of strings and corresponding Vec2 (minimum <0,0>, maximum is frame resolution) positions in the frame |
| Accuracy: | Output: More accurate than the sample program. Should perform correct identification at least 70% of the time |
| Receive/Send Frequency: | The output containing information of recognized faces will be transmitted at 30 Hz |
| Timing Issues: | Asynchronous; The application will need to use the facial recognition module to determine who is talking at any given time so that the speech recognition/construction module responds correctly. |

|  |  |
| --- | --- |
|  | |
| Module: | User Interface |
| Input: | Text and button selections |
| Output: | Config file |
| Data Type: | Output: Organized, parsable configuration data that specify how the user’s profile should be customized |
| Accuracy: | The configuration should 100% reflect the options and settings desired by the user. |
| Receive/Send Frequency: | The data will be used each time the user interacts with the application |
| Timing Issues: | During normal usage of the application, configuration data should not change. Therefore, there should not be a problem with timing the access of data by other modules. |

**3.3 Physical Environment Requirements**

|  |  |
| --- | --- |
|  | |
| Statement: | Jarvis will work on any Desktop/ Laptop as long as it is equipped with camera and microphone. Good lighting is necessary |
| Source: | Developer |
| Dependency: | None |
| Conflicts: | Low lighting will decrease the effective range/ accuracy of the system |
| Supporting Materials: | None |
| Evaluation Method: | None |
| Revision History: | Julian Rojas | 10/5/15 | Created the requirement |

**3.4 User and Human Factor Requirements**

|  |  |
| --- | --- |
|  | |
| Statement: | Any user is able to use the system. Knowledgy of supported commands is recommended in order to maximize the efficacy of program. |
| Source: | Developer |
| Dependency: | None |
| Conflicts: | None |
| Supporting Materials: | None |
| Evaluation Method: | None |
| Revision History: | Julian Rojas | 10/5/15 | Created the requirement |

**3.5 Documentation Requirements**

|  |  |
| --- | --- |
|  | |
| Statement: | A set of instructions will be included as a PDF. It will also have a list of commands. It will be available as a hard copy or on-line. |
| Source: | Developer |
| Dependency: | None |
| Conflicts: | None |
| Supporting Materials: | None |
| Evaluation Method: | None |
| Revision History: | Julian Rojas | 10/5/15 | Created the requirement |

**3.6 Data Requirements**

|  |  |
| --- | --- |
|  | |
| Statement: | Eigenanalysis algorithm is required for face recognition. Recognition requires 70% accuracy. |
| Source: | Developer |
| Dependency: | None |
| Conflicts: | Accuracy depends a lot on cam quality and lighting |
| Supporting Materials: | [Source of algorithm](http://www.codeproject.com/Articles/239849/Multiple-face-detection-and-recognition-in-real) |
| Evaluation Method: | None |
| Revision History: | Julian Rojas | 10/5/15 | Created the requirement |

|  |  |
| --- | --- |
|  | |
| Statement: | Jarvis is required to keep profiles of saved users. These include application/website preferences. |
| Source: | Developer |
| Dependency: | None |
| Conflicts: | None |
| Supporting Materials: | None |
| Evaluation Method: | Test Case 5 |
| Revision History: | Julian Rojas | 10/5/15 | Created the requirement |

**3.7 Resource Requirements**

|  |  |
| --- | --- |
|  | |
| 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 |

**3.8 Security Requirements**

|  |  |
| --- | --- |
|  | |
| 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 |

|  |  |
| --- | --- |
|  | |
| 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**

|  |  |
| --- | --- |
|  | |
| 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 10 |
| Revision History: | Manuel Gonzalez | 10/5/15 | Created the requirement |

|  |  |
| --- | --- |
|  | |
| 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: | Run the application over the course of several weeks and see how often it fails. |
| Revision History: | Manuel Gonzalez | 10/5/15 | Created the requirement |

|  |  |
| --- | --- |
|  | |
| 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 14 |
| Revision History: | Manuel Gonzalez | 10/5/15 | Created the requirement |

|  |  |
| --- | --- |
|  | |
| 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 13 |
| Revision History: | Manuel Gonzalez | 10/5/15 | Created the requirement |