**Statusphere CEN 5016, Spring, 2017**

**Software Requirements Specification v3**

Team Name: Statusphere Team Members:

* Project Owner: Kristen Wiley
* Project Manager: Brandon Jones
* Front-End Developers:
  + Amith Lakshmisha
  + Sukmin Kim
  + Sathya Narayanan Amarnath
  + Pooja Shivashankar
  + Swathi Kanthareddy Sumithra
  + Arpitha Rajanna
  + Leena Jawale
  + Pavitra Shavanand Hiremath
  + Nithyashree Kempegowda
* Back-End Developers:
  + Nishant Sahni
  + Atul John
  + Soham Ganguly
  + Somnath Saha
  + Gaurav Kataria
  + Xuejiao Yang

**Contents of this Document**

Introduction

Definition, Acronyms, and Abbreviations

Product Overview

* Assumptions
* Stakeholders
* Event Table
* Use Case Diagram
* Use Case Descriptions
* Specific Requirements
* Functional Requirements
* Interface Requirements
* Physical Environment Requirements
* Users and Human Factors Requirements
* Documentation Requirements
* Data Requirements
* Resource Requirements
* Security Requirements
* Quality Assurance Requirements (AKA nonfunctional, Quality Requirements)
* Supporting Material

**Section 1: Introduction**

Software to be Produced:

The team will be developing a system, Statusphere, for providing select social media *Influencers* with advertising opportunities; this entails: building a front-end, both a web portal and a mobile app, for influencers to select advertising *packages*; building a front-end for administrative tasks (such as accepting new influencers and adding new advertising packages); building a back-end for storing both influencer and advertiser package information; and, developing an automated acceptance system for selecting influencers.

**Definitions, Acronyms, and Abbreviations**

Influencer: A person with a high number of followers on one or more social media platforms; will also refer to an applicant that has been accepted as a member on Statusphere. See Class Diagram in Section 4 for more details.

Package: An item that an Influencer will *advertise* for in exchange for receiving the item for free; the Influencer must agree to advertise for the item in the manner indicated by the requirements associated with it. A Package will refer both to the item and the various data associated with it; see Class Diagram in Section 4 for more details.

Follower Engagement: A function of the number of followers and the aggregate number of likes that an Influencer has.

AWS: Amazon Web Services

SMP: Social Media Platform (Facebook, Twitter, Instagram, etc.)

UCD: Use-Case Diagram

**Section 2: Product Overview**

Assumptions:

* Each member in the teams have sufficient technical competency; but, might require training for some technologies.
* Each member is ready to work 20 hours a week, unless overtime is required to keep the project on schedule.
* Replacement team members are not available; although, team assignments can be changed to fit the workload.
* The duration of the project is limited to the UCF Spring 2017 semester.
* There is $5,000 available for infrastructure on Amazon Web Services.
* Sufficient Memory and Storage will be available on any infrastructure.

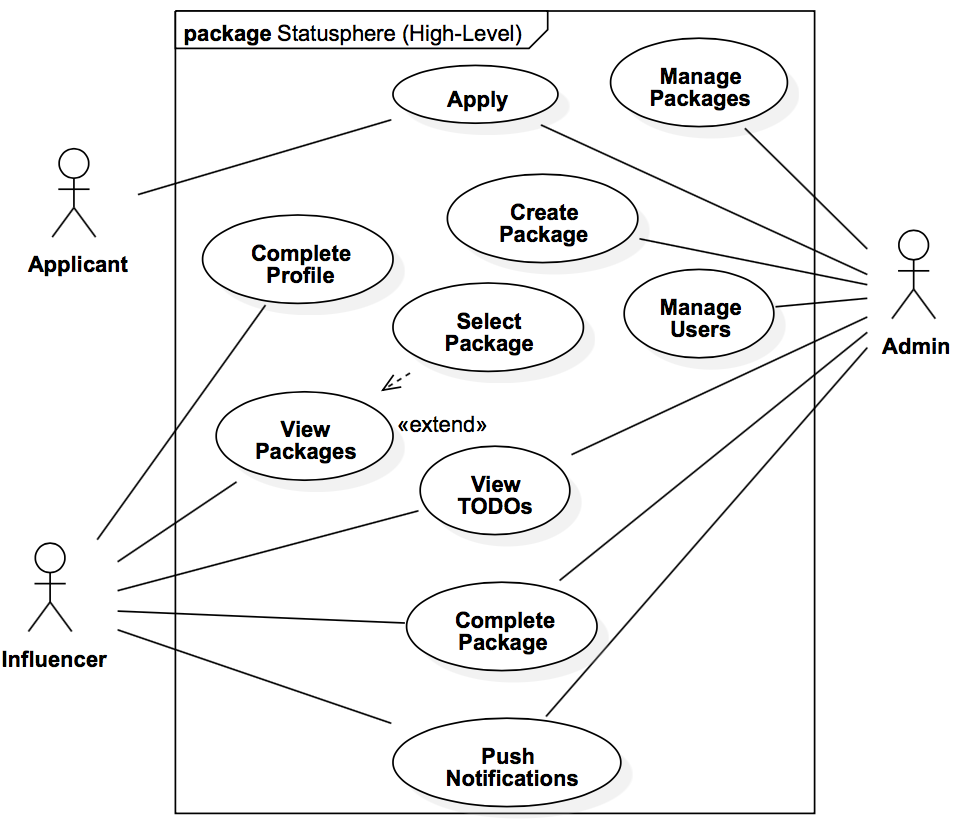
Stakeholders:

* Kristen Wiley (Joinstatus CEO): will be using the product to manage existing clients and potential new clients (applicants).
* Dr. Edwin Nassiff (Instructor): will be evaluating teams performance on the project.
* Influencers: will use the product for selecting packages and providing proof of package completion.
* Applicants: will apply to become an Influencer on Statusphere.
* Advertisers: will have their products made available for Influencers to potentially select on Statusphere; selected packages will be advertised by the Influencers on their respective social media platforms.

Event Table:

|  |  |  |  |
| --- | --- | --- | --- |
| Event Name | External Stimuli | External Responses | Internal Data and State |
| Apply for membership | User signs up the system with his username and password in any social media | The user enters his credentials with information of his activities in social media | The information is stored in the database for future analytics. |
| E-Mail for acceptance/Rejection | User is notified about his acceptance/Rejection through an email alert | User, if accepted, enters his basic, geographical and demographic information. | These information are stored for the corresponding user, so that respective product packages are assigned to user for marketing. |
| Notification regarding package readiness | The administrator  notifies the user stating the readiness of the product packages. | User checks in the packages page and selects the package assigned to him | Once selected, user is notified about the time duration within which the product should be posted in social media along with the requirement specification mentioned by administrator. |
| Complete Profile | Influencer logs into their account for the first time after being accepted | Influencer provides address and demographic information | The influencers address and demographic information are stored in the database and associated with their profile. |
| View Packages | An influencer, that has completed their profile signs in and is redirected to View Packages page | The user is able to view the details of available packages. | User’s current page is now the View Packages page. |
| Select Package | An Influencer clicks a button to add a selected package to their TODO list. | The user is thanked and returned to the package view screen. | The package is added to the Influencer’s TODO list. |
| View TODOs | An influencer clicks the TODOs tab. | User is redirected to the TODOs page | User’s current page is now the TODOs page |
| Complete Package | An influencer marks a package on their TODO list as completed | User is prompted to provide a link proving completion. | User is being prompted |
| Prove Completion | User initiated Complete Package and entered a link to proof. | Package is marked as Awaiting Verification | The user’s instance of the package is marked as awaiting verification in database, the link is uploaded to the database. |
| Begin Draft Package | An admin clicks draft package button. | The user is redirected to the package draft page. | The user is on the package entry page. |
| Complete Draft Package | The admin enters the details of the package and clicks finished. | The admin is redirected to home page. | The package is uploaded to database as a drafted package. |
| Push Packages | The admin pushes all drafted packages to relevant users. | All drafted packages are now available packages; Influencers may now view the packages; notifications are sent. | Packages flagged in database as drafted are now flagged as available; package names are sent to notification system. |
| Manage Users | Admin enters user management page | A table of users and their associated attributes is displayed. | The admin is on the user management page. |
| Filter Users | User selects an attribute to filter by and selects ranges or values to display. | Filtered users are removed from the users table. | The user management page dynamically changes. |
| Remove User | Admin selects Influencer and removes them. | Confirmation popup is displayed; if confirmed, user is removed from user table | The user is marked as removed in database (other data is kept for record keeping). |
| Place user on Probation | Admin selects user and clicks to put them on probation | User’s status now displays as on Probation | Probation flag is set to true in database for the user. |

High-Level Use-Case Diagram



Use Case Descriptions:

1. **Apply:** A potential Influencer applies for membership on Statusphere; they provide their usernames for relevant social media platforms.
2. **Complete Profile:** The user provides additional demographic and contact information necessary for selecting appropriate packages and shipping selected packages.
3. **View Packages:** An influencer views available packages which are appropriate for them (based on demographics and area of interest). Packages will typically only be available for a week (but might be available for longer).
4. **Select Package:** A user agrees to complete the requirements of the selected package.
5. **View TODOs:** The user views the list of packages which they have accepted, but have not yet been marked as completed; the admin can view uncompleted packages in tabular format (one row for each package-user pair).
6. **Create Package:** The admin uploads the details of a new package and selects the types of users that should be able to accept the packages.
7. **Push Notifications:** The Admin pushes notifications to users devices and/or email.
8. **Manage Packages:** The admin views existing packages and edits as desired (including removing a package).
9. **Manage Users:** The admin views existing users and important information associated with each (such as demographic information and number of packages completed).

For more details, refer to the Detailed Use-Case Diagram in Section 4.

**Section 3: Specific Requirements**

3.1 Functional Requirements

|  |  |
| --- | --- |
| No. | FR001 |
| Statement | The system shall provide potential users (applicants) with the ability to apply for membership on Statusphere - they must provide an email address and indicate any relevant social media platforms that they influencer on. |
| Source | Applicant |
| Dependency | None |
| Conflicts | None |
| Supporting Materials | High-Level UCD, Detailed UCD |
| Evaluation Method | Potential user applications will be made available to Kristen. |
| Revision History |  |

|  |  |
| --- | --- |
| No. | FR002 |
| Statement | The system shall allow an administrator to accept an applicant. |
| Source | Admin. |
| Dependency | FR001 |
| Conflicts | None |
| Supporting Materials | High-Level UCD, Detailed UCD |
| Evaluation Method | Accepted users will have their profile added as Influencers to the Database. |
| Revision History |  |

|  |  |
| --- | --- |
| No. | FR003 |
| Statement | The system shall allow an administrator to reject an applicant. |
| Source | Admin. |
| Dependency | FR001 |
| Conflicts | None |
| Supporting Materials | High-Level UCD, Detailed UCD |
| Evaluation Method | The associated pending application will be removed from the pending applications list. |
| Revision History |  |

|  |  |
| --- | --- |
| No. | FR004 |
| Statement | The system shall allow an Influencer to complete their profile after they have been accepted – this includes providing demographic information and indicating area(s) of interest (I.e.: Fashion, Food, etc). |
| Source | Influencer |
| Dependency | FR005, FR006 |
| Conflicts | None |
| Supporting Materials | High-Level UCD, Detailed UCD |
| Evaluation Method | Created packages will be added to the packages database. |
| Revision History |  |

|  |  |
| --- | --- |
| No. | FR005 |
| Statement | The system shall allow an Influencer to login to Statusphere using the email that they used to apply and a password that they have selected. |
| Source | Influencer |
| Dependency | FR001, FR002 |
| Conflicts | None |
| Supporting Materials | High-Level UCD, Detailed UCD |
| Evaluation Method | A user is able to enter Statusphere using valid username and password; and not, otherwise. |
| Revision History |  |

|  |  |
| --- | --- |
| No. | FR006a |
| Statement | The system shall allow an Influencer to login to Statusphere using an Instagram account. |
| Source | Influencer |
| Dependency | FR001, FR002 |
| Conflicts | None |
| Supporting Materials | High-Level UCD, Detailed UCD |
| Evaluation Method | A user is able to enter Statusphere after logging into the matching Instagram account. |
| Revision History | Separated from FR006 in version 3. |

|  |  |
| --- | --- |
| No. | FR006b |
| Statement | The system shall allow an Influencer to login to Statusphere using a Facebook account. |
| Source | Influencer |
| Dependency | FR001, FR002 |
| Conflicts | None |
| Supporting Materials | High-Level UCD, Detailed UCD |
| Evaluation Method | A user is able to enter Statusphere after logging into the matching Facebook account. |
| Revision History | Separated from FR006 in version 3. |

|  |  |
| --- | --- |
| No. | FR006c |
| Statement | The system shall allow an Influencer to login to Statusphere using a Twitter account. |
| Source | Influencer |
| Dependency | FR001, FR002 |
| Conflicts | None |
| Supporting Materials | High-Level UCD, Detailed UCD |
| Evaluation Method | A user is able to enter Statusphere after logging into the matching Twitter account. |
| Revision History | Separated from FR006 in version 3. |

|  |  |
| --- | --- |
| No. | FR006d |
| Statement | The system shall allow an Influencer to login to Statusphere using a Google+ account. |
| Source | Influencer |
| Dependency | FR001, FR002 |
| Conflicts | None |
| Supporting Materials | High-Level UCD, Detailed UCD |
| Evaluation Method | A user is able to enter Statusphere after logging into the matching Google+ account. |
| Revision History | Separated from FR006 in version 3. |

|  |  |
| --- | --- |
| No. | FR007 |
| Statement | The system shall allow an Influencer to request a password change, should they forget their password. |
| Source | Influencer |
| Dependency | FR002, FR019 |
| Conflicts | None |
| Supporting Materials | High-Level UCD, Detailed UCD |
| Evaluation Method | Indicating a forgotten password will trigger an email to an admin. |
| Revision History |  |

|  |  |
| --- | --- |
| No. | FR008 |
| Statement | The system shall allow an Influencer to view the details and requirements of packages that an Admin has made available for them to select. |
| Source | Influencer |
| Dependency | FR028 |
| Conflicts | None |
| Supporting Materials | High-Level UCD, Detailed UCD, Class Diagram |
| Evaluation Method | A view packages page shall be available to an active user. |
| Revision History |  |

|  |  |
| --- | --- |
| No. | FR009 |
| Statement | The system shall allow an Influencer to select a package which they would like to receive, in exchange for agreeing to complete the specified requirements of the package. |
| Source | Influencer |
| Dependency | FR028 |
| Conflicts |  |
| Supporting Materials | High-Level UCD, Detailed UCD, Class Diagram |
| Evaluation Method | A user shall select a package and an association between the user and the package will be created on the database. |
| Revision History | Changed requirement number from FR010 to FR009 in version 3. |

|  |  |
| --- | --- |
| No. | FR010 |
| Statement | The system shall add a package that has been accepted by an Influencer to their respective TODO list. |
| Source | Influencer |
| Dependency | FR010 |
| Conflicts | None |
| Supporting Materials | High-Level UCD, Detailed UCD |
| Evaluation Method |  |
| Revision History | Changed name from FR011 to FR010 in version 3. |

|  |  |
| --- | --- |
| No. | FR011 |
| Statement | The system shall allow an Influencer to view the list of packages that they have accepted, but not yet completed – their TODO list. |
| Source | Influencer |
| Dependency | FR010 |
| Conflicts | None |
| Supporting Materials | High-Level UCD, Detailed UCD, Class Diagram |
| Evaluation Method | The user’s TODO list will display all packages associated with the user in the database. |
| Revision History | Changed No. from FR012 to FR011 in version 3. |

|  |  |
| --- | --- |
| No. | FR012 |
| Statement | The system shall allow an Influencer to check-off requirements of a Package on their TODO list. |
| Source | Influencer |
| Dependency | FR011 |
| Conflicts | None |
| Supporting Materials |  |
| Evaluation Method | The user’s TODO list will display all packages associated with the user in the database. |
| Revision History | Added in version 3. |

|  |  |
| --- | --- |
| No. | FR013 |
| Statement | The system shall allow an Influencer to provide a link (or links) proving the completion of a package on their TODO list. |
| Source | Influencer |
| Dependency | FR012 |
| Conflicts | None |
| Supporting Materials | High-Level UCD, Detailed UCD |
| Evaluation Method | A link will be added to a user’s package instance on the database. |
| Revision History |  |

|  |  |
| --- | --- |
| No. | FR015 |
| Statement | The system shall allow an admin to push drafted notifications to any or all Influencers respective devices (if using a mobile app) or email accounts. |
| Source | Admin |
| Dependency | FR002 |
| Conflicts | None |
| Supporting Materials | High-Level UCD, Detailed UCD |
| Evaluation Method | Pushing a notification will result in an email being sent to a set of selected users. |
| Revision History | Changed language used in “Statement” and “Evaluation Method” in version 3; originally: “...to a set of *known* users” and “…push notifications…”. Changed No. to FR015 in version 3. |

|  |  |
| --- | --- |
| No. | FR017 |
| Statement | The system shall automatically push notifications to Influencers respective devices (if using mobile app) or email accounts when an Influencer has 24 hours remaining to complete a package still on their TODO list. |
| Source | Notification System |
| Dependency | FR011 |
| Conflicts | None |
| Supporting Materials | Detailed UCD |
| Evaluation Method | Packages will be created due dates slightly over 24 hours, they will be accepted by test users, and notifications should be received shortly there after. |
| Revision History |  |

|  |  |
| --- | --- |
| No. | FR018 |
| Statement | The system shall allow an admin to log into an *easy* management web portal using their email and selected password. |
| Source | Admin |
| Dependency | None |
| Conflicts | None |
| Supporting Materials | Detailed UCD |
| Evaluation Method | An admin will attempt to log into to admin portal using correct username and password. |
| Revision History |  |

|  |  |
| --- | --- |
| No. | FR019 |
| Statement | The system shall provide an admin with a web portal for viewing a table of data associated with Influencers. |
| Source | Admin |
| Dependency | FR018 |
| Conflicts | None |
| Supporting Materials | Detailed UCD, Class Diagram |
| Evaluation Method | An online admin is able to view table of users. |
| Revision History |  |

|  |  |
| --- | --- |
| No. | FR020 |
| Statement | The system shall provide an admin with a web portal for viewing a table of data associated with Packages. |
| Source | Admin |
| Dependency | FR018 |
| Conflicts | None |
| Supporting Materials | Detailed UCD, Class Diagram |
| Evaluation Method | An online admin is able to view a table of packages. |
| Revision History |  |

|  |  |
| --- | --- |
| No. | FR021 |
| Statement | The system shall provide an admin with a web portal for viewing users and associated packages. |
| Source | Admin |
| Dependency | FR018 |
| Conflicts | None |
| Supporting Materials | Detailed UCD, Class Diagram |
| Evaluation Method | An online admin is able to view a table of users and packages. |
| Revision History |  |

|  |  |
| --- | --- |
| No. | FR022 |
| Statement | The system shall allow an admin to filter the displayed rows in any displayed table by values from any of its columns. |
| Source | Admin |
| Dependency | FR019, FR020, FR021 |
| Conflicts | None |
| Supporting Materials | Detailed UCD |
| Evaluation Method | Filters will be checked for each individual combination and an assortment of popular combinations of filters. |
| Revision History |  |

|  |  |
| --- | --- |
| No. | FR023 |
| Statement | The system shall allow an admin to remove a user from the system, preventing them from logging in. |
| Source | Admin |
| Dependency | FR002, FR019 |
| Conflicts | None |
| Supporting Materials | Detailed UCD |
| Evaluation Method | A removed user will have a removed flag associated with them on the database. |
| Revision History |  |

|  |  |
| --- | --- |
| No. | FR024 |
| Statement | The system shall allow an admin to view a list of packages on any Influencers TODO list. (Requirements FR021 and FR022 can be used to together to fulfill this requirement). |
| Source | Admin |
| Dependency | FR002, FR008 |
| Conflicts | FR021, FR022 |
| Supporting Materials | Detailed-UCD |
| Evaluation Method |  |
| Revision History |  |

|  |  |
| --- | --- |
| No. | FR025 |
| Statement | The system shall allow an admin to mark a package on an Influencers TODO list as completed. |
| Source | Admin |
| Dependency | FR011, FR013, FR018 |
| Conflicts | None |
| Supporting Materials | High-Level UCD, Detailed-UCD |
| Evaluation Method | An admin will mark a package as completed for a user and that package will no longer appear on the users todo list. |
| Revision History |  |

|  |  |
| --- | --- |
| No. | FR026 |
| Statement | The system shall remove a package that has been marked by an admin from the respective Influencer's TODO list. |
| Source | Admin |
| Dependency | FR025 |
| Conflicts | None |
| Supporting Materials | High-Level UCD, Detailed-UCD |
| Evaluation Method | An admin will mark a package as completed for a user and that package will no longer appear on the users todo list. |
| Revision History | None |

|  |  |
| --- | --- |
| No. | FR027 |
| Statement | The system shall allow an admin to draft a new package, this includes: entering a description and requirements; uploading a picture (or pictures); and providing links. A drafted package is not yet available for any Influencers to select. |
| Source | Admin |
| Dependency | FR018 |
| Conflicts | None |
| Supporting Materials | Detailed-UCD |
| Evaluation Method |  |
| Revision History |  |

|  |  |
| --- | --- |
| No. | FR028 |
| Statement | The system shall allow an admin to push any drafted packages to the list of packages viewable/selectable by Influencers. |
| Source | Admin |
| Dependency | FR027 |
| Conflicts | None |
| Supporting Materials | Detailed-UCD |
| Evaluation Method |  |
| Revision History |  |

|  |  |
| --- | --- |
| No. | FR029 |
| Statement | The system shall be able to look up and display the number of followers that a user, specified by the admin, has on a platform, specified by the admin. |
| Source | Admin, Reporting Tools |
| Dependency | FR018 |
| Conflicts | None |
| Supporting Materials | Detailed-UCD |
| Evaluation Method | The details of a known user will be compared to output of system, for several different users. |
| Revision History |  |

|  |  |
| --- | --- |
| No. | FR030 |
| Statement | The system shall be able to look up and display the number of likes that a post (specified by the admin) by a user (specified by the admin) has on a platform (specified by the admin). |
| Source | Admin, Reporting Tools |
| Dependency | FR018 |
| Conflicts | None |
| Supporting Materials | Detailed-UCD |
| Evaluation Method | The details of a known user will be compared to output of system, for several different users. |
| Revision History |  |

|  |  |
| --- | --- |
| No. | FR031 |
| Statement | The system shall be able to look up and display the trimmed average number of likes per post that a user (specified by the admin) has on a platform (specified by the admin). |
| Source | Admin, Reporting Tools |
| Dependency | FR029, FR030 |
| Conflicts | None |
| Supporting Materials | Detailed-UCD |
| Evaluation Method | The details of a known user will be compared to output of system, for several different users. |
| Revision History |  |

|  |  |
| --- | --- |
| No. | FR032 |
| Statement | The system shall allow an admin to mark an Influencer as being on a probation period. Influencers on probation are not able to select new packages. |
| Source | Admin |
| Dependency | FR019 |
| Conflicts | None |
| Supporting Materials | Class Diagram |
| Evaluation Method | A user marked as being on probation will not be able to select packages. |
| Revision History | Added in version 2 as FR010; changed name to FR032 in version 3. |

3.2 Interface Requirements

|  |  |
| --- | --- |
| No. | IR001 |
| Statement | An Influencer shall be able to create an account on Statusphere by providing an email and password. The username and password shall be in alphanumeric. All the UI design shall be in English language. The information shall be stored in the database for administrator. |
| Source | Influencer |
| Dependency | None |
| Conflicts | None |
| Supporting Materials | Statusphere signup page |
| Evaluation Method | Once entered, the information is stored in the backend |
| Revision History | TBD |

|  |  |
| --- | --- |
| No. | IR001b |
| Statement | The influencer shall be able to sign up with their social media credentials to the system. The username and password shall be in alphanumeric. All the UI design shall be in English language. The information shall be stored in the database for administrator. |
| Source | Influencer |
| Dependency | None |
| Conflicts | None |
| Supporting Materials | Statusphere signup page |
| Evaluation Method | Once entered, the information is stored in the backend |
| Revision History | TBD |

|  |  |
| --- | --- |
| No. | IR002 |
| Statement | Administrator notifies the user about their acceptance based on their participation in social media applications. The notification shall be in a mail and, if using a mobile app, a push notification to the users device's push notification API. It shall be in English. |
| Source | Admin page in statusphere |
| Dependency | Depends on IR001 |
| Conflicts | None |
| Supporting Materials | Email and contact number of users |
| Evaluation Method | Confirmation on sending mail to the user |
| Revision History | TBD |

|  |  |
| --- | --- |
| No. | IR003 |
| Statement | The influencer shall fill in all of their basic and demographic information upon acceptance. All information shall be in English. Contact numbers, mail id and address may contain alphanumeric characters. |
| Source | Influencer profile in statusphere |
| Dependency | IR002 |
| Conflicts | None |
| Supporting Materials | Statusphere profile page |
| Evaluation Method | Storage of these information in the backend for future analytics |
| Revision History | TBD |

|  |  |
| --- | --- |
| No. | IR004 |
| Statement | The successful login of the user is achieved once the user logs into statusphere account with correct credentials that is with correct username and password. The information of the user is extracted from the admin database. |
| Source | Statusphere login page. |
| Dependency | IR001 |
| Conflicts | None |
| Supporting Materials | Database of users profile. |
| Evaluation Method | After logging in, the user's name with his/her account is seen to demonstrate successful login. |
| Revision History | TBD |

|  |  |
| --- | --- |
| No. | IR005 |
| Statement | The user searches the available packages on the website depending on their area of interests. The details of the packages are available on the website and the user can add the packages to the user's to-do-list. |
| Source | Statusphere website. |
| Dependency | None |
| Conflicts | None |
| Supporting Materials | Database of the available packages. |
| Evaluation Method | The packages are added to user's to-do-list only when the user agrees to complete the requirements of the selected package. |
| Revision History | TBD |

|  |  |
| --- | --- |
| No. | IR006 |
| Statement | The selected packages are available on the user's to-do-list which are not yet marked as completed. The proof of completion of a requirement is given through a social media platform. After the completion of the requirement, all the user's are notified about the new packages available to view and select. |
| Source | User's to-do-list page (in user's account) and admin portal |
| Dependency | IR005 |
| Conflicts | None |
| Supporting Materials | Database of the available packages. |
| Evaluation Method | Notifications sent regarding the packages as the proof of package availability. |
| Revision History | TBD |

|  |  |
| --- | --- |
| No. | IR007 |
| Statement | The information of each user and the information of each package is stored in the form of a table containing one row for each unique user-package pair which is accessed by the admin after logging into the admin portal. The admin removes fake users from statusphere. |
| Source | Admin portal |
| Dependency | IR001, IR006 |
| Conflicts | None |
| Supporting Materials | Database of the user profiles and packages |
| Evaluation Method | Use of data attribute to filter users from the table (statusphere) |
| Revision History | TBD |

|  |  |
| --- | --- |
| No. | IR008 |
| Statement | The remaining packages are in the user's to-do-list is viewed by the admin and the verifies the proof of completion of the package. If the package is completed, it is removed from the user's to-do-list. |
| Source | Admin portal |
| Dependency | IR005 |
| Conflicts | None |
| Supporting Materials | User's to-do-list |
| Evaluation Method | Proof of completion of package is done by removing it from the user's to-do-list. |
| Revision History | TBD |

|  |  |
| --- | --- |
| No. | IR009 |
| Statement | The details of the new packages of the user are now drafted to view and accept and the admin verifies the social media username provided to know the number of followers and average number of likes per post for that user. |
| Source | Social media platform, user's profile. |
| Dependency | IR001 |
| Conflicts | None |
| Supporting Materials | Likes for each post on the social media for the particular user. |
| Evaluation Method | A particular user is verified based on number of followers and likes for each post of that user. |
| Revision History | TBD |

|  |  |
| --- | --- |
| No. | IR010 |
| Statement | The admin shall be able to view the status of an Influencer without having to navigate away from the top-most page containing that Influencer’s name. |
| Source | Admin |
| Dependency |  |
| Conflicts | None |
| Supporting Materials | Class Diagram |
| Evaluation Method | Any and all pages pertaining to Influencers, in the Admin portal, which displays a user’s name will also display that user’s status. |
| Revision History | Added in version 2. |

3.3 Physical Environment Requirements (Nishant)

|  |  |
| --- | --- |
| No. | PR001 |
| Statement | The web portal shall work on all current versions of Safari, Chrome, and Firefox browsers; and, on iPhone 6 and iPhone 7. |
| Source | Survey results |
| Dependency | None |
| Conflicts | None |
| Supporting Materials | A survey was conducted on a sample of Statusphere users and it was determined that 10/10 used an iPhone 6 or 7. |
| Evaluation Method | The web portal shall be tested using Chrome, Firefox, and Safari browsers; the iOS app will be tested on both an iPhone 6 and 7. |
| Revision History | TBD |

3.4 User and Human Factors Requirements

|  |  |
| --- | --- |
| No. | UHF001 |
| Statement | Influencers shall be able to use all functionality of the system without any training. |
| Source | Influencer |
| Dependency | None |
| Conflicts | None |
| Supporting Materials |  |
| Evaluation Method | Test users will be observed performing various tasks with the system. |
| Revision History | TBD |

|  |  |
| --- | --- |
| No. | UHF002 |
| Statement | Admins shall be able to use all functionality of the system with only limited training. |
| Source | Influencer |
| Dependency | None |
| Conflicts | None |
| Supporting Materials |  |
| Evaluation Method | An admin will be observed performing various tasks with the system while being given the opportunity to ask any usage questions. |
| Revision History | TBD |

3.5 Documentation Requirements

|  |  |
| --- | --- |
| No. | DOC001 |
| Statement | The team shall document all changes in requirements and design. |
| Source | CEN 5016 |
| Dependency | None |
| Conflicts | None |
| Supporting Materials |  |
| Evaluation Method | The team will be graded on the quality of various documentation deliverables. |
| Revision History | TBD |

|  |  |
| --- | --- |
| No. | DOC002 |
| Statement | The team shall provide a user’s manual, describing step-by-step how to perform any action that a user should be able to do (see Event Table); the manual will cover 3 user types: Applicants, Influencers, and Administrators. |
| Source | CEN 5016 |
| Dependency | None |
| Conflicts | None |
| Supporting Materials | Event Table |
| Evaluation Method | The team will be graded on the quality of various documentation deliverables. |
| Revision History | Added in version 2 |

|  |  |
| --- | --- |
| No. | DOC003 |
| Statement | The team shall track changes to major documentation deliverables using a version control system (such as Git) |
| Source | CEN 5016 |
| Dependency | None |
| Conflicts | None |
| Supporting Materials | https://github.com/bwj-GitHub/CEN5016-Statusphere |
| Evaluation Method | Statement of Work, Project Management Plan, Software Requirement Specification (this document) and Software Design Description documents will be tracked on GitHub. |
| Revision History | Added in version 2 |

3.6 Data Requirements

|  |  |
| --- | --- |
| No. | DR001 |
| Statement | The system shall collect Influencers’ data including official Instagram, Facebook, Twitter, Snapchat handles. |
| Source | Influencer |
| Dependency | DR002 |
| Conflicts | None |
| Supporting Materials | <https://developers.facebook.com/>  <https://www.instagram.com/developer/>  <http://snaplytics.io/> |
| Evaluation Method | If we are able to login through all necessary social media outlets, the requirement is met. |
| Revision History | TBD |

|  |  |
| --- | --- |
| No. | DR002 |
| Statement | The system shall collect Influencers’ likes, share, retweets, hashtags, audience interaction data. This will be used to fuel analytics |
| Source | Influencer |
| Dependency | None |
| Conflicts | None |
| Supporting Materials | <https://developers.facebook.com/>  <https://www.instagram.com/developer/>  <http://snaplytics.io/> |
| Evaluation Method | If all information analytics is received, requirement is met. |
| Revision History | TBD |

|  |  |
| --- | --- |
| No. | DR003 |
| Statement | Determine the area of influence and demographic of the Influencer |
| Source | Influencer |
| Dependency | None |
| Conflicts | None |
| Supporting Materials | None |
| Evaluation Method | If area of influence and demographic is available for all Influencers, requirement is met |
| Revision History | TBD |

3.7 Resource Requirements

|  |  |
| --- | --- |
| No. | RR001 |
| Statement | Personnel shall be having skills of HTML, CSS, Bootstrap to develop front-end, Python , Django and Mongo-DB to develop back-end and Swift to develop IOS app and there shall be enough software, laptops and IPhones for development. |
| Source | Various tutorial websites and websites to download all the software. IPhone for IOS Developers and Individual Laptops. |
| Dependency | None |
| Conflicts | None |
| Supporting Materials | None |
| Evaluation Method | If personnel having skills with required software and hardware will be able to develop their respective parts successfully, the requirement will be satisfied. |
| Revision History | TBD |

3.8 Security Requirements

|  |  |
| --- | --- |
| No. | S001 |
| Statement | The system shall only store user passwords in the database in hashed format. |
| Source | Security |
| Dependency | FR005, FR018 |
| Conflicts | None |
| Supporting Materials | None |
| Evaluation Method |  |
| Revision History | Added in version 2 |

|  |  |
| --- | --- |
| No. | S002 |
| Statement | The system shall ensure that a user’s personally identifiable information (name, address, demographic information) is only able to be accessed through web (and mobile) portals by that user and any administrators (that are logged in). |
| Source | Security |
| Dependency | FR004, FR005, FR018 |
| Conflicts | None |
| Supporting Materials |  |
| Evaluation Method | Web requests for resources containing PII should include a security token identifying that user. |
| Revision History | Added in version 2 |

Note: money is not handled in the system; also, by using “Login with [Social Media Platform]” options, third-party sites will be responsible for maintaining the security of the user’s login credentials.

TODOs:

* Plan to keep the backup of data depending upon its volume (TBD).
* Plan to backup all data in a separate machine (TBD).
* Recovery requirement (TBD)

3.9 Quality Assurance Requirements (AKA nonfunctional, Quality Requirements)

1. Usability:

|  |  |
| --- | --- |
| No. | U001 |
| Statement | The system shall provide a uniform look and feel between any and all web pages mobile applications. |
| Source | Admin |
| Dependency | None |
| Conflicts | None |
| Supporting Materials | None |
| Evaluation Method | Test users will evaluate using an 11-point Scale. |
| Revision History |  |

|  |  |
| --- | --- |
| No. | U002 |
| Statement | The system should be able to interact with multiple popular SMPs. |
| Source | Admin |
| Dependency | FR006, FR029, FR030 |
| Conflicts | None |
| Supporting Materials | None |
| Evaluation Method | The system will, at least partially, implement several of the following APIs: Facebook, Instagram, Twitter, and YouTube. |
| Revision History |  |

|  |  |
| --- | --- |
| No. | U003 |
| Statement | The system should be available to Influencers as a iOS mobile application. |
| Source | Influencers |
| Dependency | None |
| Conflicts | None |
| Supporting Materials | None |
| Evaluation Method | An iOS app will be available on the Apple Store. |
| Revision History |  |

1. Performance:

|  |  |
| --- | --- |
| No. | P001 |
| Statement | The system should be able to support at least 10 concurrent users (Applicants, Influencers, and Administrators) without a notable degradation of performance. |
| Source | System |
| Dependency | None |
| Conflicts | None |
| Supporting Materials | None |
| Evaluation Method | Concurrent user tests (performed by developers); performance will be evaluated qualitatively. |
| Revision History | Added in version 2. |

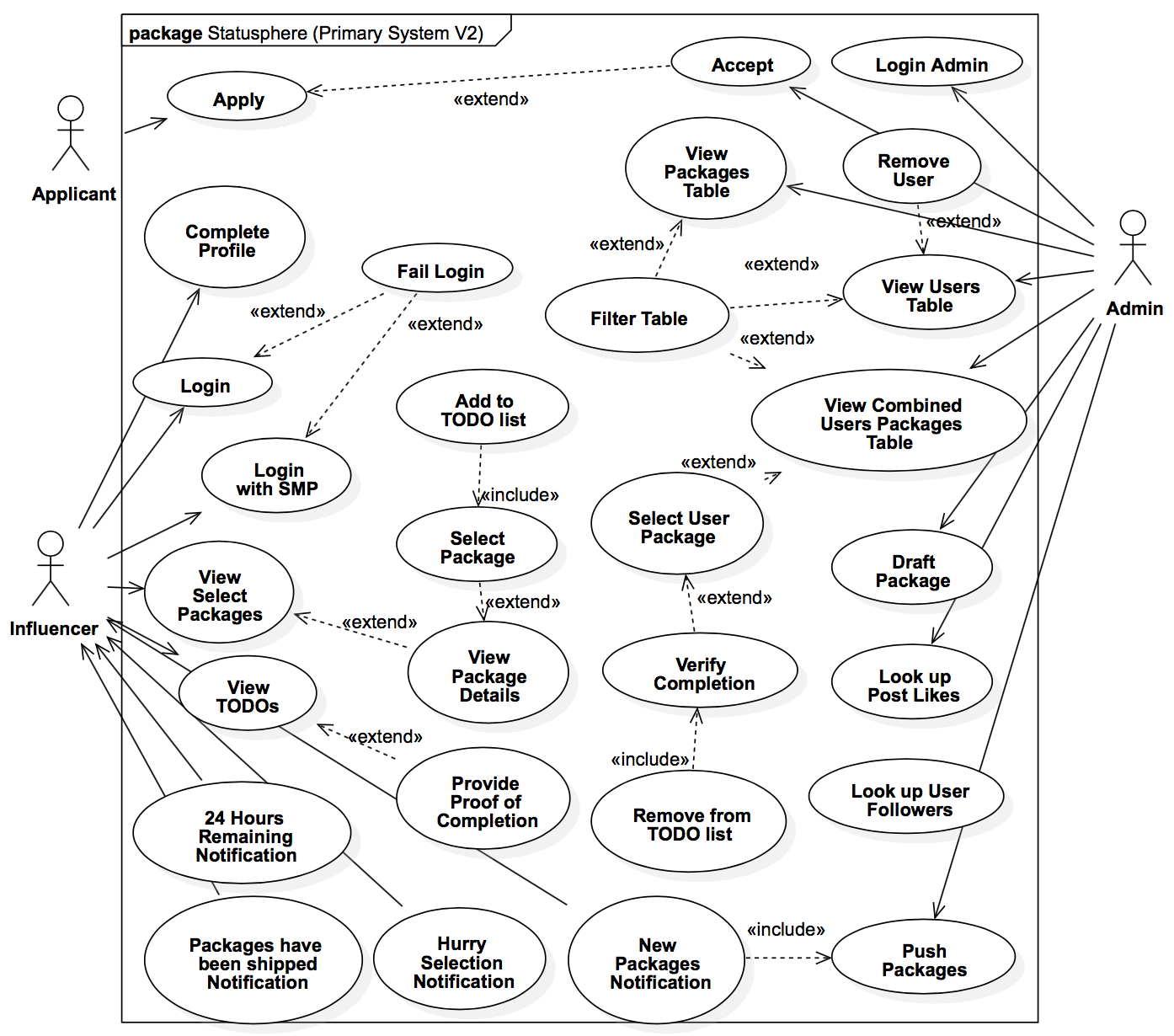
|  |  |
| --- | --- |
| No. | P002 |
| Statement | The system should allow all actions to be performed in an amount of time acceptable to most users and to Kristen. |
| Source | System |
| Dependency | None |
| Conflicts | None |
| Supporting Materials | None |
| Evaluation Method | Test users will evaluate system usability using an 11-point scale. Aside from loading packages to view, no actions on the Influencer portal should require more than 2 seconds wait time (preferably no more than 1 second). Loading Package View page (all selectable packages) should not take more 10 seconds, given internet speeds of at least 1MBPS. |
| Revision History | Added in version 2. |

1. Reliability and Availability:

There are currently no specific requirements regarding reliability or availability.

**Section 4: Supporting Material**

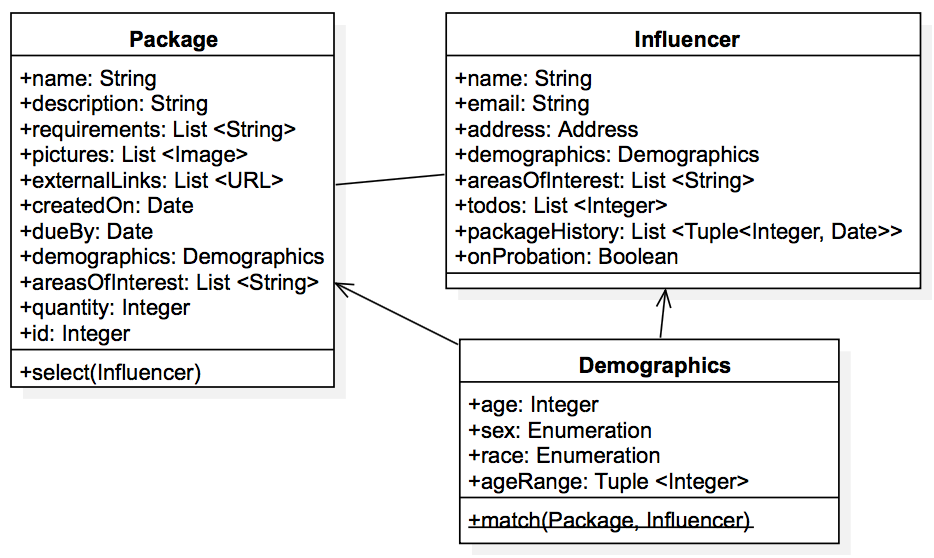
Detailed Use-Case Diagram:



Detailed Use-Case Descriptions:

1. **Apply:** A potential Influencer applies for membership on Statusphere; they provide their usernames for relevant social media platforms.
2. **Accept:** An administrator (or the System) determines that the applicant has sufficient follower engagement to be accepted as a member on Statusphere – they are added to the system as an Influencer.
3. **Reject:** An administrator rejects an applicant.
4. **Complete Profile:** The user provides additional demographic and contact information necessary for selecting appropriate packages and shipping selected packages.
5. **Login:** The user logs into their Statusphere account using their email and Statusphere password. An influencer must log into their account before they can perform any other actions.
6. **Login with SMP:** The user logs into their Statusphere account using their Twitter or Facebook account.
7. **Fail Login:** An influencer does not provide the correct credentials to log into their account.
8. **View Select Packages:** An influencer views available packages which are appropriate for them (based on demographics and area of interest). Packages will typically only be available for a week (but might be available for longer).
9. **View Package Details:** The user views detailed information about a package.
10. **Select Package:** A user agrees to complete the requirements of the selected package.
11. **Add to TODO List:** The selected package is added to the user's to-do list.
12. **View TODOs:** The user views the list of packages which they have accepted, but have not yet been marked as completed; the admin can view the list of
13. **Provide Proof of Completion:** The user provides a link to where proof of the completion of a requirement can be found (this link should be to an appropriate social media platform).
14. **Push New Packages Notification:** The Admin pushes notifications to all users devices that new packages are available for them to view and select (if there is at least one package available to them).
15. **Packages have been Shipped Notification:** An influencer is notified that a package or packages have been shipped to them.
16. **Hurry Selection Notification:** An influencer is notified that they only have 24 hours remaining to select available packages.
17. **24 Hours Remaining Notification:** An influencer is notified that they have not yet completed the requirements of a package (or the proof has not yet been verified) and only have 24 hours remaining to do so.
18. **Login Admin:** The admin logs in.
19. **View Users Table:** The admin views a table with the data associated with each user.
20. **View Packages Table:** The admin views a table with the data associated with each Package.
21. **View Combined User Packages Table:** The admin views a table of users combined with their selected packages (the table contains one row for each unique User-Package pair).
22. **Filter Table:** The admin filters a table by some data attribute (any data attribute can be filtered on).
23. **Remove User:** The admin selects a user from the list of members and removes them from Statusphere.
24. **View Outstanding Packages:** The administrator views a list of the packages remaining on a user's to-do list
25. **Verify Completion:** The administrator examines the link that a user provided as proof of completion of a package and determines that the requirement has been fulfilled
26. **Remove from TODO list:** The completed package is removed from the user's to-do list.
27. **Draft Package:** An administrator fills in the details of a new package, including to what sorts of users the package should be made available to – the package is NOT yet made available to users.
28. **Push Packages:** An administrator makes drafted packages available for users to view and potentially accept.
29. **Look up User Followers:** An administrator provides a username for a social media platform and is provided with the number of followers that that user has on the platform.
30. **Look up Post Likes:** An administrator provides a username and a social media platform and is provided with the (trimmed) average number of likes per post for that user.

Class Diagram:



Application Sequence Diagram:

