**Skoovy Android App**

Hot Salsa Interactive LLC

Cole, Tyler; Kowal, Anthony; Matthews, Lilith; Wever, Rudi

The purpose of the Skoovy app is to deliver easy access to crowdsourced text and/or media geotagged content about locations, services, and events to Skoovy users.

By using this app, users can see details about interesting places, see the quality of food at a restaurant, and see events happening at desired locations.

One primary reason solving this problem is important can be understood by its implementation. Services currently exist that provide information on the same topics that Skoovy does, but they are largely text based with occasional photographs. Video can provide so much more information with greater detail. To further understand why solving this problem is important, it is necessary to break the issue down into the three categories Skoovy focuses on: Locations, Food, and Events. The first category, locations, will allow users to show off things like a store’s offering or the atmosphere of a party venue. Food posts will show the quality of meals served at restaurants. Video will also allow for users to verbalize their feelings after a meal. The last category is ~~a~~ special as it is a time sensitive category. Events are posts about specific things happening in the area. Some examples could be a parade, a party, or even a riot or protest.

Skoovy will provide users a quick and easy way to create and share videos regarding the previously mentioned categories. Beyond this, it will also provide a quick and intuitive way for finding this content. To facilitate the creation of content that will be useful to users, a layer of gamification is placed on top of the service. Users will have points that they can spend to ask questions regarding an area. These questions can be seen and answered by other users, and upon answering they are rewarded with more points. If a post is found to be particularly liked by many users, this will also provide points. All users will publicly see points, thus the desire to have a large point count will drive users to create quality content and answer many questions.

**Requirements and Schedule**

**REQ 1. User authentication**

|  |  |  |
| --- | --- | --- |
| **Task** | **Due date** | **Status** |
| User table in database | January 28 | **new** |
| GUI for login and password request | January 28 | **new** |
| Create credentials for valid users | January 28 | **new** |
| Authenticator component – it does the authentication | January 28 | **new** |
| GUI for failure on authentication | January 28 | **new** |

**REQ 2. Camera Function**

|  |  |  |
| --- | --- | --- |
| **Task** | **Due date** | **Status** |
| Access device camera for photos/videos | March 3 | **new** |
| Add layers to photos/videos for FX | March 3 | **new** |
| Add layers for text | March 3 | **new** |
| Add geolocation | March 3 | **new** |
| Make content public or private | March 3 | **new** |

**REQ 3. Create Map View**

|  |  |  |
| --- | --- | --- |
| **Task** | **Due date** | **Status** |
| Show a current map to user | February 28 | **new** |
| Show content to user | February 28 | **new** |
| Pinpoint map based on user GPS | February 28 | **new** |

**REQ 4. User Profile**

|  |  |  |
| --- | --- | --- |
| **Task** | **Due date** | **Status** |
| Tie user profile to user login | February 10 | **new** |
| Create profile requirements | February 10 | **new** |

**REQ 5. User Requests**

|  |  |  |
| --- | --- | --- |
| **Task** | **Due date** | **Status** |
| Create messaging/notification system | February 22 | **new** |
| Create points system for managing amount of requests | February 22 | **new** |
| Tie messaging/points system to user profile | February 22 | **new** |

**REQ 6. Algorithms to decide what content to show on a list view or map view**

|  |  |  |
| --- | --- | --- |
| **Task** | **Due date** | **Status** |
| Format the table of results | March 21 | **new** |
| Create algorithm | March 21 | **new** |
| Create options to change algorithm | March 21 | **new** |

# Software Design

## Class Diagram

## 

## Relational Database Diagram

