Team 4

arizona state university | ser 401 – senior project

Skoovy

Software Design SPECIFICATIONS

**Table of Contents**

[1. Introduction 1](#_Toc166398920)

[1.1. Document Outline 2](#_Toc166398921)

[1.2. Document Description 4](#_Toc166398922)

[1.2.1. Introduction 4](#_Toc166398923)

[1.2.2. System Overview 5](#_Toc166398924)

[2. Design Considerations 5](#_Toc166398925)

[2.1. Assumptions and Dependencies 5](#_Toc166398926)

[2.2. General Constraints 5](#_Toc166398927)

[2.3. Goals and Guidelines 6](#_Toc166398928)

[2.4. Development Methods 6](#_Toc166398929)

[3. Architectural Strategies 6](#_Toc166398930)

[4. System Architecture 7](#_Toc166398931)

[4.1. Subsystem Architecture 8](#_Toc166398932)

[5. Policies and Tactics 8](#_Toc166398933)

[6. Detailed System Design 9](#_Toc166398934)

[6.1. Classification 9](#_Toc166398935)

[6.2. Definition 9](#_Toc166398936)

[6.3. Responsibilities 10](#_Toc166398937)

[6.4. Constraints 10](#_Toc166398938)

[6.5. Composition 10](#_Toc166398939)

[6.6. Uses/Interactions 10](#_Toc166398940)

[6.7. Resources 10](#_Toc166398941)

[6.8. Processing 10](#_Toc166398942)

[6.9. Interface/Exports 11](#_Toc166398943)

[6.10. Detailed Subsystem Design 11](#_Toc166398944)

[7. Glossary 11](#_Toc166398945)

[8. Bibliography 11](#_Toc166398946)

*NO CHANGES MADE FROM DRAFT. REVISION WILL BE DONE BY FIRST WEEK OF SPRING CLASSES AS PER INSTRUCTOR ANNOUNCEMENT.*

1. Introduction
   1. Document Outline

Here is the outline of the proposed template for software design specifications. Please note that many parts of the document may be extracted automatically from other sources and/or may be contained in other, smaller documents. What follows is just one suggested outline format to use when attempting to present the architecture and design of the entire system as one single document. This by no means implies that it literally is a single document (that would not be my personal preference):

Introduction

System Overview

Design Considerations

Assumptions and Dependencies

General Constraints

Goals and Guidelines

Development Methods

Architectural Strategies

strategy-1 name or description

strategy-2 name or description

...

System Architecture

component-1 name or description

component-2 name or description

...

Policies and Tactics

policy/tactic-1 name or description

policy/tactic-2 name or description

...

Detailed System Design

module-1 name or description

module-2 name or description

...

Glossary

Bibliography

* 1. Document Description
     1. *Introduction*

Provide an overview of the entire document:

* Describe the purpose of this document
* Describe the scope of this document
* Describe this document's intended audience
* Identify the system/product using any applicable names and/or version numbers.
* Provide references for any other pertinent documents such as:
  + Related and/or companion documents
  + Prerequisite documents
  + Documents which provide background and/or context for this document
  + Documents that result from this document (e.g. a test plan or a development plan)
* Define any important terms, acronyms, or abbreviations
* Summarize (or give an abstract for) the contents of this document.

Note:

For the remaining sections of this document, it is conceivable (and perhaps even desirable) that one or more of the section topics are discussed in a separate design document within the project. For each section where such a document exists, a reference to the appropriate design document is all that is necessary. All such external (or fragmented) design documents should probably be provided with this document at any design reviews.

* + 1. *System Overview*

1. Design Considerations
   1. *Assumptions and Dependencies*

Skoovy users (Skoovers) will need to utilize a mobile device in order to use Skoovy, such as a smart phone or tablet. Skoovers mobile devices will need to have a camera, data connection, and a GPS signal and they will need to allow the app access to those aspects of their device. The app will be available on Android devices.

End users will be those who use social media networks frequently, most likely users who are accustomed to Snap Chat and Instagram. The majority of end users will be aged 15-30, female, and in the US.

Functionality may change based on changes in common hardware of mobile devices. Probable changes include changes in the “social” aspect of Skoovy, including messaging and chat functions. Possible changes include implementing a “Featured videos” and meet up function, as well as a feed based on locations.

* 1. *General Constraints*

1. Platform

A major constraint that the project will face is its platform. The system shall be developed using the Android architecture and delivered to Android phones. As a direct result of this, we will need to ensure that our app shall be able to run on a multitude of Android products, specifically phones and tablets. Our app will need to be able to run on these platforms without any detectable decreases in performance. Depending on the complexity of the design, this will rule out older devices from being compatible with the app. However, with the trend of upgrading devices every one to two years, it is not anticipated that this will be a concern.

1. Hardware

There are several system requirements of the device that need to be considered. By design the app assumes the device it is running on has a camera, a touchscreen, a microphone, and an internet connection at minimum. The app is also expected to send and video data to and from a server, so spotty or poor internet connectivity will have a detrimental effect to the app’s utility.

1. Backend

This app is intended to be the front-end for a social service and will be expected to handle user’s information in an ethical and legal manner. Safeguards will need to be implemented to ensure the safety of the customer’s information. There will also need to be work ensuring that customers can access their data easily and securely through authentication methods.

* 1. *Goals and Guidelines*

1. Apply The KISS principle ("Keep it simple and straightforward!").

The eight requirements that identify for a good design which are well structured, simple, efficient, adequate, flexible, practical, implementable and standardized are the guidelines to create this design.

1. Emphasis is on speed versus memory use on this app.
2. Working, looking, or "feeling" like an existing app.

The goal of this project is to deliver the product completed on time. Use all the recommended models in the design document during coding. The app shall have the feel of a mix between the current apps Waze, Uber, Snapchat. The look shall be very simple and intuitive just as the aforementioned apps. The app is intended to be a GPS based, crowd-sourced, social networking app. The core concept revolves around making short 30 -second videos about interesting things in a certain geolocation. Examples include food at a restaurant or a protest in another country. Point based system incentivizes posts against requests. All profiles have points, unless starting with a new account. Points are gained by making posts. Points can be spent to make requests for posts at certain areas. When an user requests information about a place, they lose a point, when answering a question by providing content to the site they gain a point. The end goal is creating a map of videos highlighting interesting information about locations. The goal of the Skoovy app is to generate revenue by providing a marketplace. In apps like Snapchat, Snow, and Line, marketplaces exist where users can buy filters to apply to their photos and videos. They also have “stickers” that are images that can be sent to other users in messages.

* 1. *Development Methods*

The Skoovy project will employee Feature Driven Development (FDD), a variant of agile methodology. With the Skoovy Project Charter document and the Skoovy Software Requirements Specification document, an overall model of the Skoovy system has been established. Working in a series of two-week “build by feature” iterations, the project team will build small but useful features for the client. Based on a core set of software engineering best practices, FDD aims at providing client/stakeholder-valued results.

1. Architectural Strategies
2. Use of a particular type of product (programming language, database, library, etc. ...)

There will be a SQL Server 2005 database involved in this system. The programming language will be Java for android app only.

1. Reuse of existing software components to implement various parts/features of the system

For additional features re-use of the forms is possible.

1. Future plans for extending or enhancing the software

This Skoovy app is a basic prototype. More additional features can be added if needed.

1. User interface paradigms (or system input and output models)

All users with a account will be able to post media as well as view it.

1. Error detection and recovery

Error detection and recovery will be done. To be able to separate error-handling code from the regular code, we will add exception errors in the code.

1. External databases and/or data storage management and persistence

External databases are involved in the code and will be used to store media of the posts.

1. Communication mechanisms

TCP/IP network communication is required as this application involves network and internet connectivity to connect to the Skoovy online tool.

1. Management of other resources

The only additional resource that needs to be managed is the internet and network resources. The connectivity of the network and internet needs to be checked frequently and throw error messages if there is any connection problems.

1. System Architecture
   1. *Subsystem Architecture*
2. Policies and Tactics
3. Choice of Product & Language

The project will use Android Studio for development as it has multiple tools to facilitate the quick and easy construction of Android apps. The project will be Java based, and will use the built-in compiler to build the app.

1. Coding conventions

Code that is easily read, understood, and loosely coupled can be maintained easier and provide more allowances for faster evolution and future improvements. Refactoring will take place early and often to facilitate in generating quality code. Comments shall also be used to ensure future programmers understand the design and function of the code. This will include a block comment before every method explaining the use of the method.

1. Development methodologies and techniques

To further facilitate the generation of quality code, several programing methodologies and techniques will be implemented. Test Driven Development will ensure that code meets all requirements. SCRUM will also be used to ensure speedy delivery of features.

1. Testing Plans

Unit Tests will be constructed for most methods to ensure that they operate correctly. This will be especially true for any methods that implement other methods.

1. Interface

The app's interface design is 100% touch based. The key interfaces of the app will be maps, video players, user comments, and video capture. Design precedent for these types of interfaces should be followed to ensure ease of use and quick learning.

1. Detailed System Design
   1. *Classification*

The VideoPlayer module is a subsystem of the application that makes up a quarter of the apps functionality. It is one of the more important modules as it provides a core feature in the form of content delivery.

* 1. *Definition*

The VideoPlayer module is capable of video playback as well as displaying user comments and ratings.

* 1. *Responsibilities*

The VideoPlayer module will primarily be responsible for playing the videos recorded by users and providing social tools, like comments and ratings, on top of the video.

* 1. *Constraints*

The VideoPlayer module can only function when a connection to the server is present as it requires the downloading of user comments and video data. The downloaded data will be cashed in a temporary folder, invisible to the user.

* 1. *Composition*

A description of the use and meaning of the subcomponents that are a part of this component.

* 1. *Uses/Interactions*

All other modules will interact with the VideoPlayer module in some manner. Results generated by the Search module will be played in the VideoPlayer module. Posts found in the Map module will be played in the VideoPlayer module. The Camera module will use a slightly modified version of VideoPlayer module to playback recorded content before uploading.

* 1. *Resources*

The VideoPlayer module will primarily read video files and user data, but will also be responsible for writing new user data in the form of comments and ratings.

* 1. *Processing*

The VideoPlayer module should use only a small amount of processing to play back the video files.

* 1. *Interface/Exports*
  2. *Detailed Subsystem Design*

1. Glossary

***Baseline***. A baseline is a work product that has been formally reviewed and accepted by the involved parties. A baseline is changed only through formal configuration management procedures.

***Protocol***. A set of conventions that govern the interaction of processes, devices, and other components within a system.

***Task***. The smallest unit of work subject to management accountability.

**Acronyms:**

**ASU** is Arizona State University.

**FK** is Foreign Key.

**FX** is Filter Effects.

**GPS** is Global Positioning System.

**ID** is Identification.

**jpg** is a photo file format developed by Joint Photographic Experts Group.

**LAMP** is an archetypal model of web service [solution stacks](https://en.wikipedia.org/wiki/Solution_stack), named as an [acronym](https://en.wikipedia.org/wiki/Acronym) of the names of its original four [open-source](https://en.wikipedia.org/wiki/Open-source) components: the [Linux](https://en.wikipedia.org/wiki/Linux) [operating system](https://en.wikipedia.org/wiki/Operating_system), the [Apache HTTP Server](https://en.wikipedia.org/wiki/Apache_HTTP_Server), the [**MySQL**](https://en.wikipedia.org/wiki/MySQL) [relational database management system](https://en.wikipedia.org/wiki/Relational_database_management_system).

**mp4** is a video file format developed Moving Pictures Experts Group.

**MTBF** is Mean Time Between Failures.

**PIN** is Personal Identification Number.

**PK** is Primary Key.

**SER** is Software Engineering.

**Skoovers** is the plural of a Skoovy user.

**SMS** is Short Message Service, also known as a text message.

**UI** is User Interface.

1. Bibliography

Reference: <http://www.agilemodeling.com/essays/fdd.htm>