6-7-2021

StickARs: Effortlessly Apply a Fun Overlay to the Real World

Théo Arrouye

Jackson Centeno

Morgan Fleshren

Vasilis Odysseos

SANTA CLARA UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Date: June 7, 2021

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY

Théo Arrouye
Jackson Centeno
Morgan Fleshren
Vasilis Odysseos

ENTITLED

StickARs: Effortlessly Apply a Fun Overlay to the Real World

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING

Thesis Advisor

DocuSigned by:
Nam Ling
96CE94A1A83A48A...

Department Chair

StickARs: Effortlessly Apply a Fun Overlay to the Real World

by

Théo Arrouye
Jackson Centeno
Morgan Fleshren
Vasilis Odysseos

Submitted in partial fulfillment of the requirements
for the degree of
Bachelor of Science in Computer Science and Engineering
School of Engineering
Santa Clara University

Santa Clara, California
June 7, 2021

StickARs: Effortlessly Apply a Fun Overlay to the Real World

Théo Arrouye
Jackson Centeno
Morgan Fleshren
Vasilis Odysseos

Department of Computer Science and Engineering
Santa Clara University
June 7, 2021

ABSTRACT

Augmented Reality (AR) technology has slowly seeped into the mainstream as many mobile phone manufacturers increasingly develop hardware and software that enable AR applications. While the technology has been available for the past few years, there are not many compelling AR applications available on mobile phones. Additionally, shared AR experiences are difficult to produce and share. In order to create a compelling AR application, we created StickARs, an iOS application that makes it easy to create and share AR experiences , by placing virtual stickers anywhere in the world.

StickARs allows users to choose from a predefined set of sticker templates, add text to stickers, and place them share them with the public, or with specific friends. Users can opt to add tags to their stickers, and subscribe to tags to see stickers that interest them.

The greatest obstacle we faced was our ability to create a world-scale shared experience. We split the entire world into equal sized chunks called worldmaps that users instantiate the first time they place a sticker at that location. This allows users to only download the worldmaps they actually visit. However due to geolocation and AR limitations, the transition between worldmaps is not quite as smooth as we had originally hoped. We hope that in the future, Apple will extend the functionality of their AR Application Programming Interface to allow larger-scale experiences, which will allow us to create a smoother shared AR experience.

ACKNOWLEDGMENTS

We would like to thank our advisor, Dr. Ahmed Amer, for his continued encouragement and support.

We would also like to thank the School of Engineering for providing us with the necessary funding to complete our project.

Table of Contents

List of Figures

Chapter 1

Introduction

This chapter discusses the overall idea behind our Senior Design problem, the problems that led us to come up with the idea, and our proposed solution to those problems.

1.1 Problem Statement

Augmented Reality (AR) promises to enrich our world by embedding it with valuable information, yet a lack of compelling AR products are available to the public. One reason for this is that dedicated AR hardware is not yet mature and mainstream, and the experience is somewhat less compelling when you must hold a screen between yourself and the world. Another issue is that AR experiences take a lot of effort to produce and share with the public. We are hoping that by solving the latter problem, we may create a compelling software product for the first generation of widely accessible consumer AR hardware.

1.2 Project Purpose

We are developing a platform for the creation and sharing of virtual stickers – using AR and geolocation to display them in real-world locations. This platform is built on Apple's ARKit, meaning it requires an iOS Device. We are operating under the assumption that Apple will launch AR glasses in the near future. These glasses would presumably use ARKit (and likely Swift/SwiftUI) - thus our work for the iOS version should provide a solid foundation for eventually deploying our platform on these glasses. This hardware is integral to our vision, as the platform is less compelling when you are separated from the content. Whenever applicable, we attempted to build our platform such that it is abstracted from ARKit, so that it could be ported to other platforms in the future.

1.3 Project Proposal

Through the StickARs iOS app (and eventually AR glasses), users can place, view, and share virtual stickers placed around the world. When a user wants to view or place stickers, they must first relocalize into a worldmap. A worldmap is a square section of the world that contains stickers for that area. Worldmaps have different levels of visibility: public, private, or shared with specific users. To place a sticker, the user must first select a template from our preset template library, and then they can add custom text and tags. Users can follow tags and/or people to customize what appears in their feed. Users can move throughout the world, and as they exit one worldmap they will be prompted to relocalize in the new worldmap. Once the user relocalizes successfully, they can enjoy the various stickers in the new worldmap.

Chapter 2

General Description of Project

In this chapter we list out all of the functional and non-functional requirements that we came up with at the beginning of our project. Requirements are deemed functional if they are necessary for the project to be "done." Non-functional requirements are the superfluous elements that we would like to implement at a later stage. Requirements marked with an asterisk (*) have been implemented.

2.1 Functional Requirements

- AR object functionality*
  - The StickARs app will have an AR view so the user can view the stickers placed around them in AR.
- 2D sticker models*
- 3D sticker models*
- AR sticker map reloads with stickers in same spot*
  - A user can close the app and return at a later time with the stickers still in the same place.
- AR sticker maps are public/shared*
  - A user can choose to make their sticker public so that any other StickARs user can view and enjoy it.
- User authentication*
  - All users must log into the app so that we can moderate users.
- The user can place a sticker*
- The user can adjust the orientation of the sticker*
- The user can block another user from their view
- The user can delete their sticker*

2.2 Non-Functional Requirements

- The user can add text to a sticker*

- The user can post stickers to a private worldmap*
- The user can disable follow requests*
- The user can submit a personal sticker to be used
  - Users can design their own stickers to use within the app. This requires the user to submit their sticker for a review process.
- The user can add tags to a sticker*
  - The user can tag a sticker with topics that represent it. For example, a mushroom sticker denoting wild mushrooms might be tagged with #mushroom.
- The user can follow tags*
- The user can join groups
  - Users can share stickers with other users having similar interests.
- The user can follow people*
  - A user can choose to follow their friends or people who they are interested in.
- The user can share a specific sticker
  - A user can share a sticker outside of the StickArs app. This link will open to the sticker's information page within the app. This page states the sticker's location, tags, and creator.
- The user can adjust the size of the sticker
- The user can set a self-delete timer when sticker is created
  - If a user wants to post a sticker for a certain time period they can select that at the time they create the sticker.
- The user can report offensive stickers
- Sticker models contain audio
  - When the user selects the sticker it begins playing audio.
- The user can interact with a sticker
  - Sticker can contain a link or a video placed in AR.
- Restricted areas/objects
  - Certain places such as monuments and cemeteries should not allow placement of stickers.
- Advertisers can post AR 2D or 3D model advertisements
  - Advertisers can pay us to place a virtual ad on our app.
- StickARs system can recommend popular stickers in your area

2.3 Design Constraints

- StickARs is an app on the App Store
  - Requires an Apple device to access the App Store.
- Requires iOS 14.0 or later
  - We are using SwiftUI 2.0 and ARKit 4.0, new APIs which are only available starting with version 14.0 of iOS.

Chapter 3

Use Cases

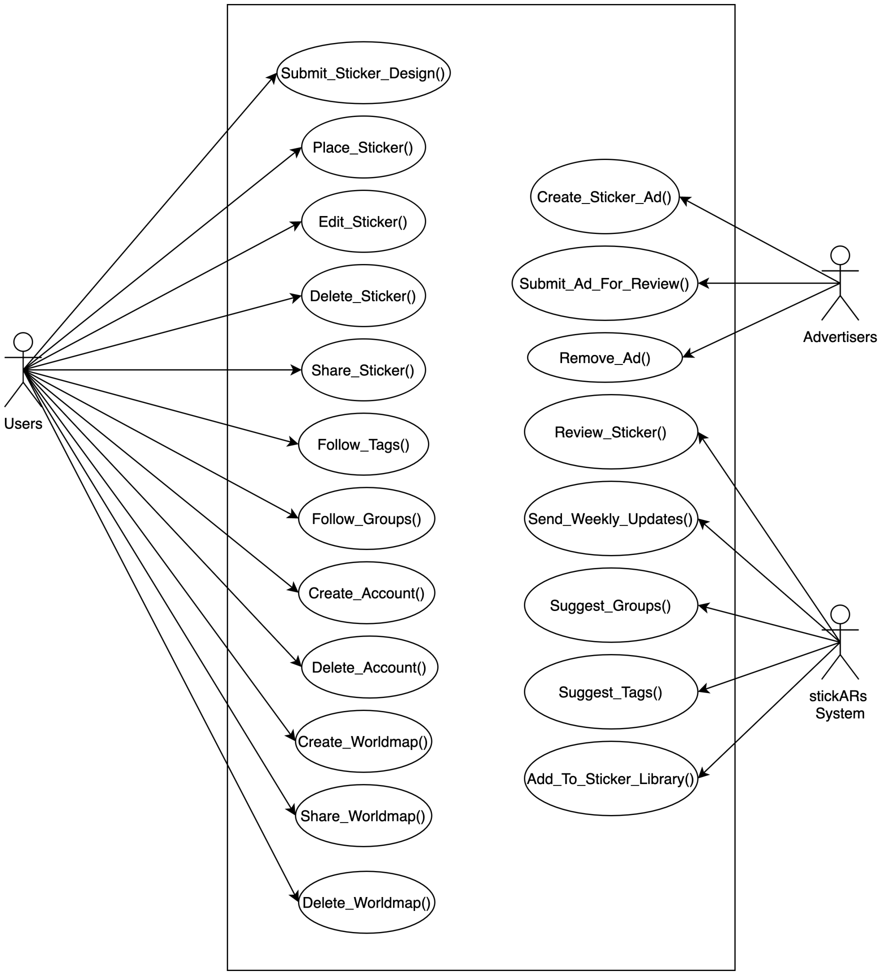


Figure 3.1: Use Case Diagram

Figure 3.1 identifies the actors in our system and the actions that each of them can perform. Our three actors are: Users, Advertisers, and the StickARs system. Each of these actors has their own set of tasks that they perform for the application.

3.1 User

This section describes the tasks performed by the User using StickARs.

3.1.1 Submit_Sticker_Design()

In this method, the user submits a sticker template that they designed. Users wanting to use their own designs must first submit the designs to StickARs so that we can moderate what gets turned into a sticker.

3.1.2 Place_Sticker()

This method is performed by the user aiming their device at a desired location and tapping to place a sticker. Then, they can drag their sticker to make fine adjustments.

3.1.3 Edit_Sticker()

This method allows the user to make their sticker unique. Users can add custom text and tags to their stickers.

3.1.4 Delete_Sticker()

This method is performed when the user decides they want to remove their sticker from the platform.

3.1.5 Share_Sticker()

Since StickARs has a social media aspect, it is important that individual stickers can be easily shared with friends. Users can click on a sticker and select to share a link containing information about the sticker, as well as the sticker itself.

3.1.6 Follow_Tag()

Users can choose to follow one or more tags in order to improve their experience. By following a tag, a user can look at stickers around them that align with their interests.

3.1.7 Follow_Group()

Similar to the tags, a user can follow a group that aligns with their interests. In a group, people can share stickers they found or made.

3.1.8 Create_Account()

In order to use the StickARs app, users must first create their account. Creating an account consists of the users submitting information such as their name, birthday, email address, and a desired username.

3.1.9 Delete_Account()

At any point in time a user can decide to remove their account from StickARs, resulting in any of their stickers also getting deleted from the StickARs platform.

3.1.10 Create_Worldmap()

The world is divided into Worldmaps: digital representations of 100mx100m areas. When the user first visits a new place, they will have to create a worldmap to begin adding stickers. Users can also create private worldmaps, where they can invite their friends and share stickers privately.

3.1.11 Delete_Private_Worldmap()

If the user decides they don't want a specific Worldmap any longer, they can delete it, which also deletes all stickers saved in that Worldmap.

3.2 StickARs System

This section describes the tasks performed by the StickARs System.

3.2.1 Send_Weekly_Updates()

The StickARs system will send updates, if the user opts in, about new stickers in their area, popular stickers in their area, new stickers under tags they follow, new stickers in groups they follow, etc.

3.2.2 Suggest_Groups()

StickARs will suggest popular groups to the user to improve their experience. The more groups they follow, the more tailored their experience is.

3.2.3 Suggest_Tags()

StickARs will also suggest tags to users to improve the experience.

3.2.4 Review_Sticker()

StickARs will review any personal sticker submissions as well as advertisement sticker submissions prior to allowing the sticker to exist on our app.

3.2.5 Add_To_Sticker_Library()

When a sticker successfully passes the review process, it will be added to the sticker library for personal use.

3.3 Advertisers

This section describes the potential tasks performed by Advertisers on StickARs.

3.3.1 Create_Sticker_Ad()

Advertisers must first create their ad in the form of a sticker to be used on our app.

3.3.2 Submit_Ad_For_Review()

Before an advertisement can be placed in the StickARs worldview or made available in the publicly accessible stickers library, the advertiser must submit their sticker through a review process.

3.3.3 Remove_Ad()

At any point the advertiser may choose to pull their advertisement from the application.

Chapter 4

User Stories

In this chapter we list out a variety of ways users might end up using our platform. Our user stories include some for individual users as well as businesses. These two are separate because while we would like to also offer our technology to businesses as a framework for integrating AR experiences into their business model, this would require additional functionality to be built out.

4.1 Individual User

4.1.1 Street Art

A user can submit their personalized stickers to StickARs for review, and then place their stickers and tag them to gain exposure. Other users will see these stickers if they follow any of the tags and can choose to follow the artist.

4.1.2 Group Treasure Hunt

A user can create a private event by placing their stickers (or the "clues") in a worldmap shared with specific users. Only the selected users can see and participate in this event.

4.1.3 Social Media

Users can choose to follow certain people and tags. This is to enhance the user's experience and let them customize what they see in their feed. While exploring for new stickers, users can find and follow new users with similar interests.

4.1.4 Private Events

The user can submit their own stickers to be used within the app. This can be a permanent sticker or a timed-deletion sticker. To use their own sticker, the user will have to pay a fee and submit the sticker beforehand. Personalized stickers require a review by the StickARs team.

4.1.5 Sticky Notes

The user can post sticky note stickers around their work area. This may be a note with their passwords, a to-do list, or simply motivational quotes.

4.1.6 Mushroom Hunting

Users can place mushroom stickers where they find mushrooms worth noting. For mushroom hunters, they could post stickers denoting which mushrooms are edible and which mushrooms are potentially dangerous. For example, user that follows #MorelMushrooms would see when a new sticker is posted, and they could navigate towards the sticker to collect some of those tasty mushrooms.

4.2 Business (as a framework)

4.2.1 Guided Tour

The user can walk in an AR guided tour. The user will see arrows on the floor to direct them, as well as any AR objects relevant to the tour content. This can allow anyone to take a tour without needing a human guide or knowledge of a certain language. This can be used in museums as well as towns with many tourists. This also helps to eliminate tour crowds and allow for social distancing due to COVID-19.

4.2.2 Golf Course

A golf course can publish a “cheat sheet” to beat the course. The golf course will place stickers marking where (notated by an x) and how hard to hit the ball (notated by the color of the x) in an effort to play the “perfect” game. Users can pay a fee to the golf course to gain access.

4.2.3 Hiking Trails

Park rangers can go through trails and make virtual trail markers. Sometimes trails are not taken care of and trail markers can get damaged/taken down. Virtual trail markers are much easier for the hiker to see and read since a sticker does not get weathered or hidden by overgrown plants. Hikers can also place stickers for their own use. A hiker going down an unmarked trail might place their own stickers so they do not get lost (or so others can find them if they do get lost/hurt).

4.2.4 AirBnB Instructions

The owner of an AirBnB or other rental property can set up a private map and send a link to the renters. Stickers they may place include sticky notes indicating where to find towels or a billboard with the rental rules.

4.2.5 Rock Climbing

Rock climbers who have completed the course (or the business owning the rock wall) can place stickers on the rocks denoting which rocks are good to grab (hand sticker) and which rocks make good footing (foot sticker).

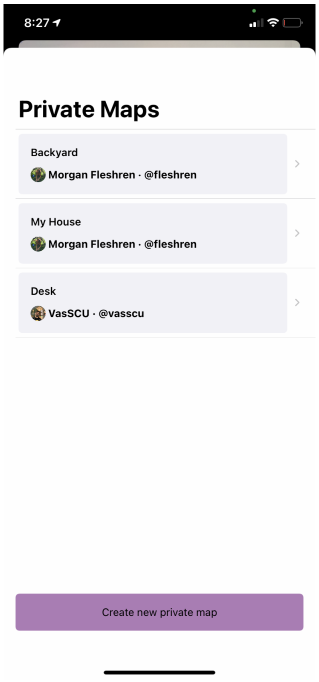
Chapter 5

Conceptual Model

This chapter contains real screenshots from our app, along with detailed explanations of what each screenshot is. Later in the chapter we also have a state diagram, context diagram, and class diagram. These diagrams can be used to gain a more in-depth understanding of how our app is structured.

5.1 UI Screenshots

5.1.1 Main Tab - AR View



Private Maps

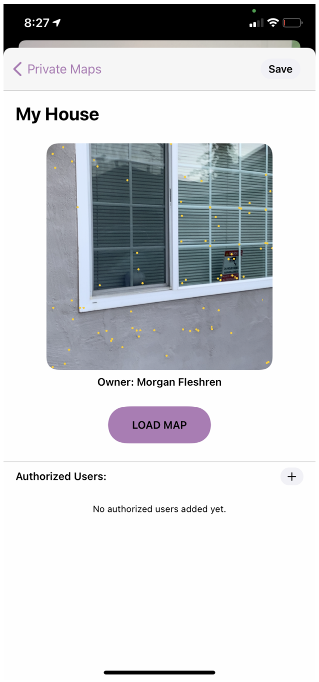
Backyard
Morgan Fleshren · @fleshren

My House
Morgan Fleshren · @fleshren

Desk
VasSCU · @vasscu

Figure 5.1: Private Map List

Users can choose to view stickers on a private or public map. Public maps are open to all users; private maps are open to specific authorized users.



My House

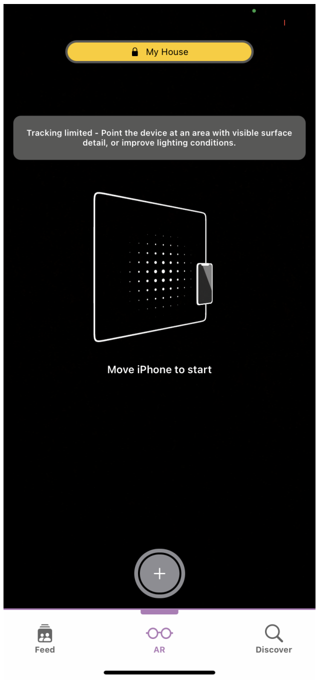
Owner: Morgan Fleshren

Authorized Users:

No authorized users added yet.

Figure 5.2: Private Map Info View

In order to load the map, the user must relocalize at the spot in the photo. We suggest using a permanent structure for the best results.



Tracking limited - Point the device at an area with visible surface detail, or improve lighting conditions.

Move iPhone to start

Figure 5.3: Relocalization Guide

Users will see this message if their phone is not picking up enough points on various surfaces. The app needs a well-lit area to work best.

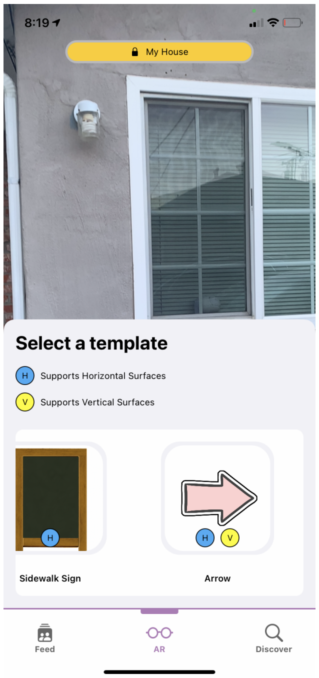


Figure 5.4: Sticker Template Library

Once the user has tracked enough points, they can choose a sticker to place. We have a preset library of sticker templates for users to choose from.

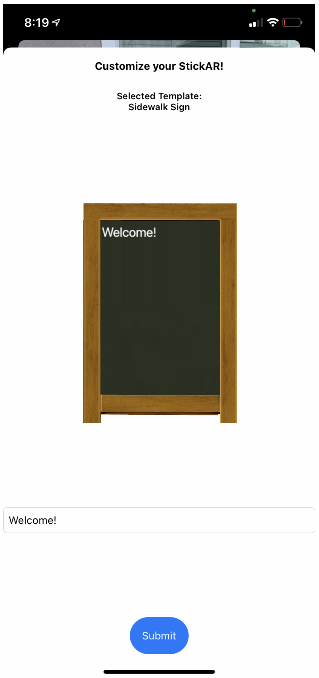
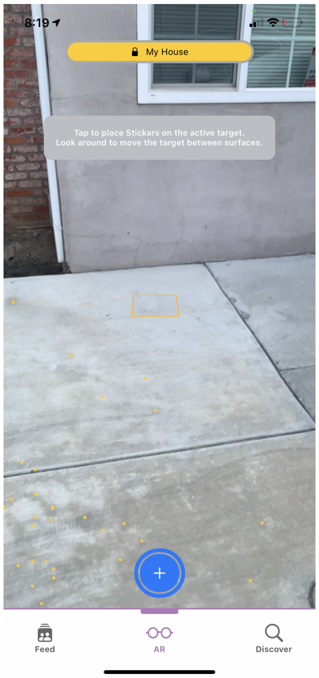


Figure 5.5: Sticker Editing View

Users can add text to their sticker before they place it. In later versions of the app, users will be able to change the text font, size, and color.



Tap to place Stickers on the active target.
Look around to move the target between surfaces.

Figure 5.6: Sticker Placement

Each yellow dot is used to identify flat surfaces for placing stickers. The square represents where the user can place a sticker.

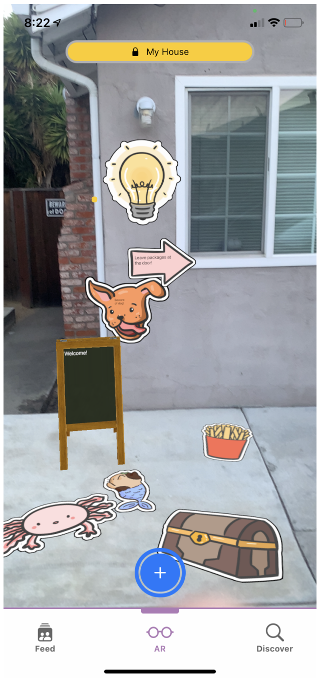
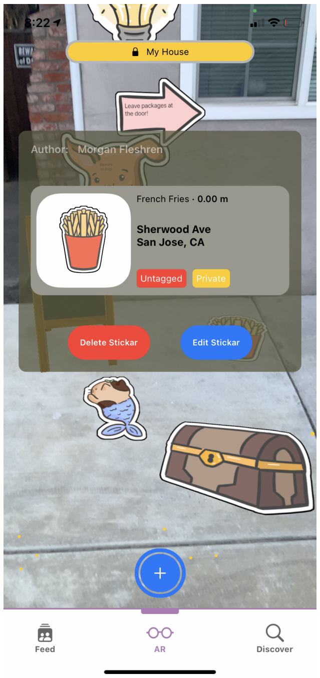


Figure 5.7: Sticker Examples

Here is a fun example of some of the stickers available in the app.



Author: Morgan Fleshren

French Fries · 0.00 m

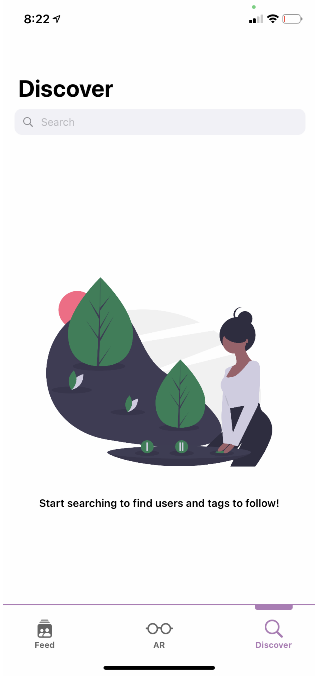
Sherwood Ave
San Jose, CA

Untagged Private

Figure 5.8: Sticker Info View

Tapping on a sticker reveals more information about it. If you are the sticker's author, you can edit or delete it.

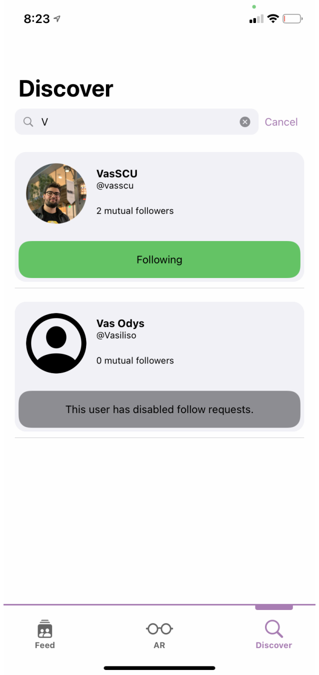
5.1.2 Discover Tab



Start searching to find users and tags to follow!

Figure 5.9: Discover Tab

In the Discover Tab, users are prompted to search for people or tags to follow.



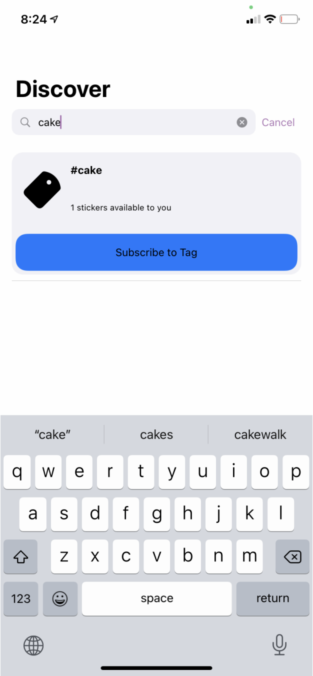
2 mutual followers

0 mutual followers

This user has disabled follow requests.

Figure 5.10: Search for Users

In this search, we found two users that start with V. One of the users has turned off their follow requests so their activity can remain private.

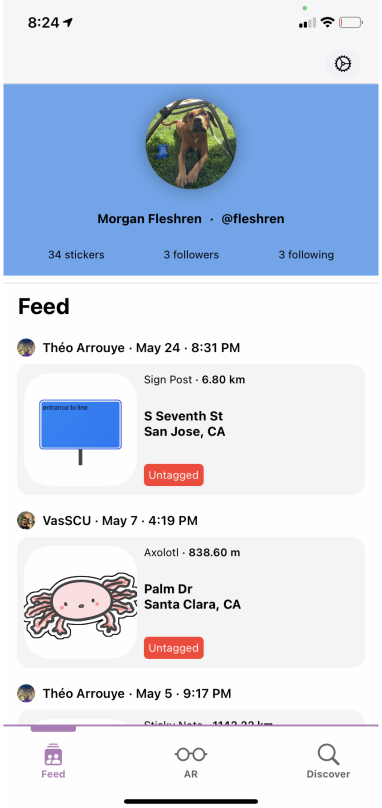


1 stickers available to you

Figure 5.11: Search for Tags

Users can also search for tags. In this search, we found a tag for #cake.

5.1.3 Feed Tab



8:24

Morgan Fleshren · @fleshren

34 stickers 3 followers 3 following

Feed

Théo Arrouye · May 24 · 8:31 PM

Sign Post · 6.80 km

S Seventh St
San Jose, CA

Untagged

VasSCU · May 7 · 4:19 PM

Axolotl · 838.60 m

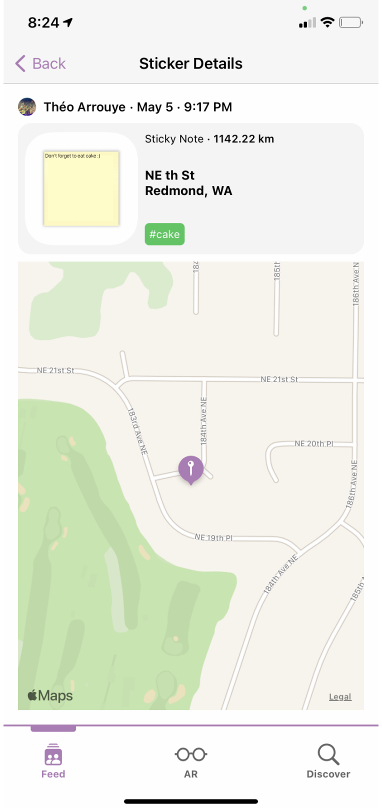
Palm Dr
Santa Clara, CA

Untagged

Théo Arrouye · May 5 · 9:17 PM

Figure 5.12: Feed View

On the user's feed there is a chronological list of stickers that have been posted by either people or tags you follow. It lists the type of sticker, the distance from the sticker to the user, and any tags.



8:24

Back Sticker Details

Théo Arrouye · May 5 · 9:17 PM

Sticky Note · 1142.22 km

NE th St
Redmond, WA

#cake

Figure 5.13: Sticker Info in Feed

If the user clicks on a sticker from their feed they can see more information about the sticker. With this map, the user can navigate to the exact location of the sticker to see it in person.

5.2 State Diagram

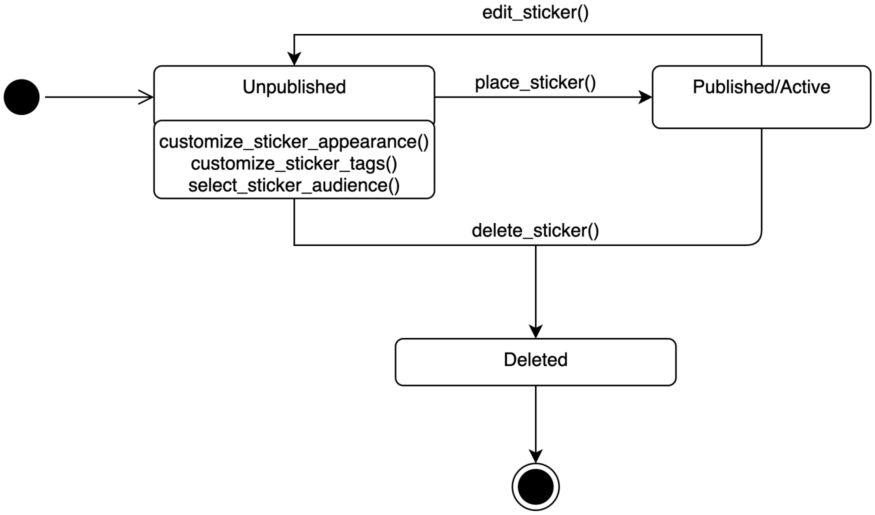


Figure 5.14: Sticker State Diagram.

The life of a sticker has 3 states. All stickers begin in the state "Unpublished." In this state, stickers are customized in terms of their color, tags, groups. The user can also select whether to publish this sticker for others to see or strictly for private use. From this state the user can take two actions: delete_sticker() or place_sticker(). They can delete_sticker() which will move the sticker to the "Deleted" state. In this state the sticker is no longer visible. The user can also place_sticker() from the "Unpublished" state. This will result in the sticker state changing to "Published." From the "Published" state, the user has two options: edit_sticker() and delete_sticker(). The edit_sticker() function returns the sticker to the "Unpublished" state where the user can choose to make additional customizations to the sticker. To republish the sticker, they need to place_sticker() again. If the user selects to delete_sticker() from any state, the sticker moves to the "Deleted" state and reaches the end of its lifetime.

5.3 Context Diagram

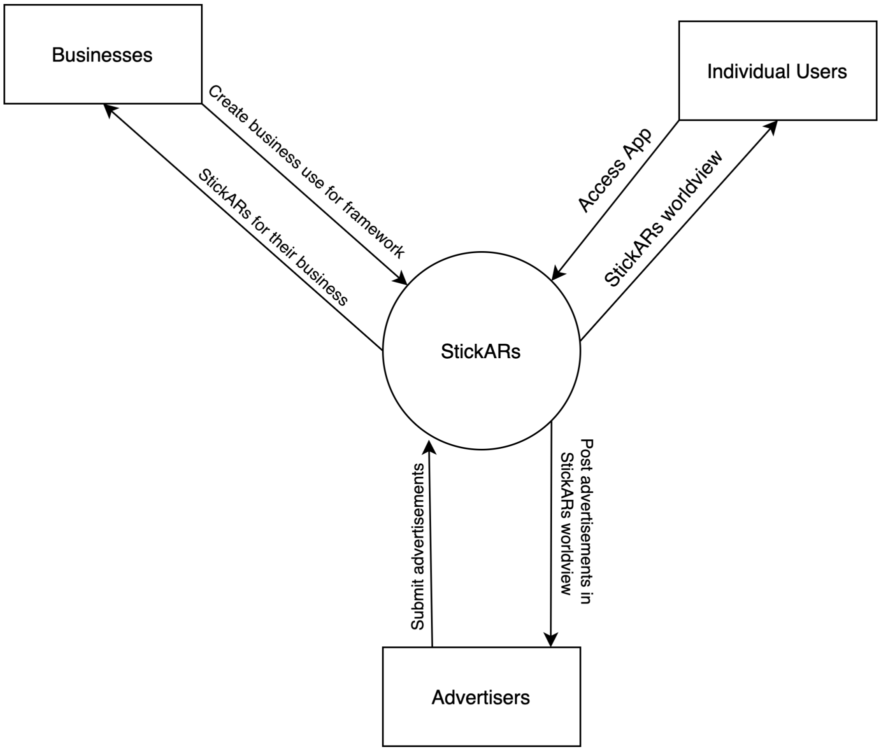

Businesses
Individual Users
Create business use for framework
StickARs for their business
Access App
StickARs worldview
StickARs
Submit advertisements
Post advertisements in StickARs worldview
Advertisers

Figure 5.15: Context Diagram.

Figure 5.15 outlines the contexts in which various stakeholders interact with StickARs. The three stakeholders we identified here are: individual users, businesses. and advertisers. Individual users interact with the app by accessing it, and in return StickARs gives them the StickARs worldview. Businesses ideate and create a use for our framework within their business, and in return we host the StickARs worldview for their business – separate from the publicly accessible StickARs worldview. Advertisers create and send us their sticker advertisements, and in return we allow their advertisements to be placed within the publicly-accessible StickARs worldview.

5.4 Class Diagram

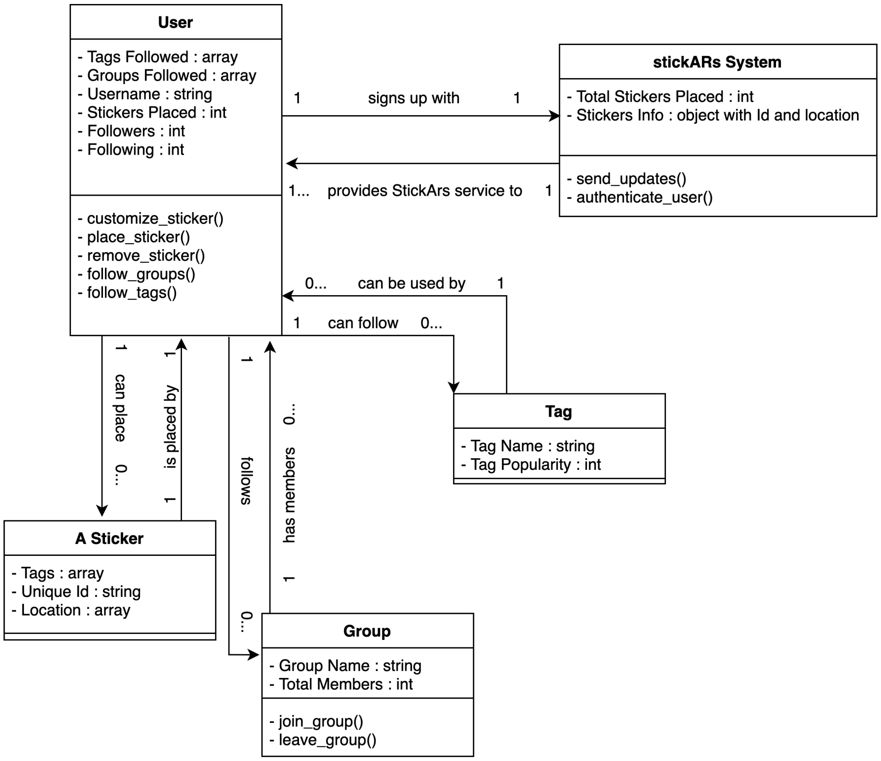


Figure 5.16: Class Diagram

Figure 5.16 defines the various classes within our application. We have six classes: User, Account, StickARs System, Tag, Group, and Sticker. For every one user, there is only one StickARs system that they will sign up with. For every one StickARs system, one or more users are provided with StickARs service. The class made between the user and the StickARs system is the user's account. Each user has one account in which personal information is stored.

The user also interacts with tags. One user can follow zero or more tags. Tags are an optional element, but they will improve the user's experience if they choose to use it. Since a tag does not get stored until a user uses that tag on a sticker, for every one tag there will be one or more users using it on stickers.

Similar to tags, the user interacts with the optional class, Group. A single user can join zero or more groups (again, this is an optional element to improve their experience). A single group requires one or more members within it to exist.

The last class in our application is the sticker itself. Each user can place zero or more stickers. It is not a requirement to place any stickers to maintain an account with StickARs. For each sticker placed in the worldview, there is exactly one person who customized and placed it.

Chapter 6

Architectural Diagram

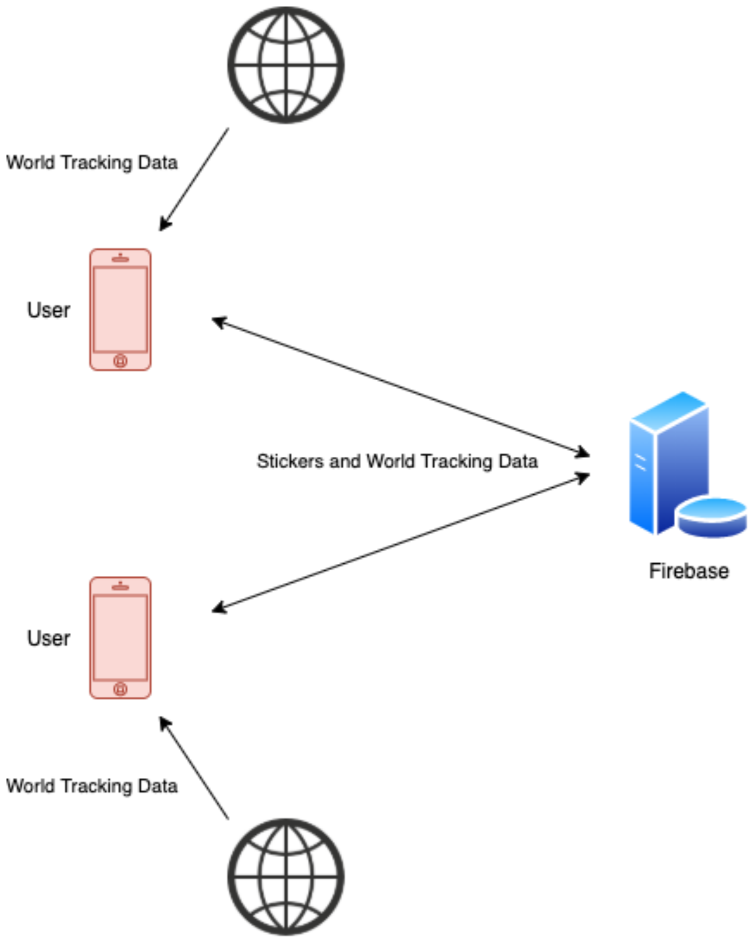


Figure 6.1: Architectural Diagram

Figure 6.1 shows how StickARs loads and stores data from the Firebase server. When StickARs is launched, world tracking data and stickers, if available, are loaded onto the user's phone from the Firebase server. As the user adds stickers onto new real world surfaces, the sticker information is uploaded to the server immediately, and the gathered world tracking information is sent to the server to be used by other users in the area.

Chapter 7

Design Rationale

In this chapter we talk about the major design choices we made for the app and why we made each choice. This includes: which view the app opens up to, the preset sticker library, an external database, and a few other choices.

7.1 Opening to the AR View

When the User first opens our application, they are immediately presented with the AR View and instructions on how to establish proper world-tracking (see Figure 5.1.1.A). This serves two purposes:

1. It gets world-tracking established as quickly as possible so that stickers loaded from the cloud can be displayed.

2. It signals to the user that this is the main way they will interact with the application.

7.2 Customizable Templates

Because some users might not be capable of designing a 2D or 3D model compatible with the app, we decided to provide a library of stickers for the user to select from. Users can customize the text on the sticker template, and in the future we hope to add the ability to customize colors. This design allows us to offer choice and customization to the user while limiting the potential for offensive or inappropriate stickers to be created.

7.3 Focus Target

The iOS Application uses an interface element we refer to as the ‘focus target.’ This yellow square is displayed along detected surfaces when the user is in placement mode. As the User moves their device around, the focus target moves along detected surfaces such that it is always in the center of their view if possible. This signifies to the user the current state of tracking along the surface, and shows them where their sticker will place

7.4 Background saving and loading

All placed stickers are automatically uploaded to the cloud if connection allows. If this fails, the user will be alerted that the upload will be re-attempted when the app reconnects to the internet. When the server receives new stickers, they are sent to other devices to be displayed where appropriate. At no point does the user ever explicitly save or load stickers from the cloud. This is to help preserve the feeling that stickers are simply a part of the world around the user.

7.5 Private Sessions

The main use-case for our application demands that the user have a reliable internet connection while using the app. However, we envision other use-cases where the user may never want to share any of the stickers they create, and an internet connection would be superfluous (e.g. setting up a virtual scene to film a video for social media). To facilitate these use-cases and allow users with privacy concerns or poor internet access to use our application in creative ways, we offer Private Sessions which store all data locally. If the user launches the app without a reliable connection, the application will offer to start or load a Private Session.

7.6 External Database

We store stickers and user info in an external database, thus allowing us to enable a shared experience between multiple users whose devices all sync to this external database. Using an external database also saves the user more space on their device. We did, however, implement caching to lessen the amount of reads/writes that are done to the database.

Chapter 8

Technologies Used

This chapter discusses the various technologies we employ to bring StickARs to life. These technologies include programming languages, frameworks, Application Programming Interfaces, and tools required to build and deploy our application.

8.1 Swift / SwiftUI

Swift is an open-source compiled programming language developed primarily by Apple. It has replaced Objective-C as the native language for Apple's platforms such as iOS and macOS.

Since we are developing a native iOS Application, Swift was the natural choice of language. Our iOS app is 100% Swift code. Swift provides strong type-safety and other helpful compile-time features which help prevent writing faulty code.

Our interface is also built entirely using SwiftUI 2.0. SwiftUI was introduced in 2019 as a new method for developing interfaces in applications on Apple's platform's. We chose SwiftUI over UIKit because it is a newer technology and we believe that in its current form it is capable of powering the vast majority of our interface. Where necessary, we dip back into UIKit.

8.2 ARKit

ARKit is an Apple-provided SDK for Augmented Reality experiences. ARKit makes use of the plethora of sensors aboard an iOS device to capture information about the world around it. Utilizing ARKit allows us to piggy-back on the tremendous work that Apple has done developing algorithms that can detect planes in the real-world. Using these detected planes, we are then able to allow placement of virtual content along real-world surfaces.

8.3 Xcode

Xcode is Apple's IDE and is required for building and publishing applications for macOS, iOS, and other Apple platforms. It runs exclusively on macOS. Since we are developing an iOS application, we had no choice but to use Xcode. It provides features such as integration with git source control, integration with Apple's code-signing tools, and integration with iTunes Connect for app distribution. It also allows us to test our application in the iOS Simulator or on a connected iOS device.

8.4 Firebase

Firebase is a platform provided by Google which offers a plethora of features for developing the backend of your applications. Firebase provides tools for tracking analytics, reporting app crashes, authenticating users, storing data, and more. We plan to use Firebase for authentication of users, as well as for storage of user data such as placed stickers, groups they've joined, etc.

8.5 Github

Github is a platform that offers hosting and version control for software development, built upon Git. We are using Github to facilitate working as a team on a shared codebase.

8.6 SceneKit

SceneKit is a 3D graphics API provided by Apple on their platforms. It is a higher-level framework which provides an easier-to-use API than lower-level graphics APIs such as OpenGL and Metal. We use SceneKit in our iOS application for the rendering of all 3D content such as the virtual stickers users place.

8.7 Clubhouse.io

Clubhouse is a project-management tool that is specifically oriented towards software development teams. As such, it felt particularly well-suited to our team and project. Clubhouse provides integration with Github so that we can update project goals and tasks remotely via actions such as marking commits or merging branches.

8.8 Testflight

Testflight is an app that allows us to have beta testers. We must first upload our app to Testflight, then our testers can download the app and submit bug reports through Testflight. Users can find our app through searching Testflight, or by following a unique link for downloading that we send them directly.

Chapter 9

Timeline

This chapter outlines our project's development timeline and possible further development after Senior year.

9.1 Summer 2020

- May: Team Formed; Project Area Decided (Augmented Reality)
- June: Brainstorming
- July: Project Decided
- August: Identified Requirements and User Stories

9.2 Fall 2020

- Oct 9th: Problem Statement Document Finished
- Oct 15th: App UI Prototype Built
- Oct 23rd: Undergraduate Funding Request Submitted
- Nov 1st: First Draft of UML Diagrams
- Nov 5th: AR Sticker Placement / Editing Prototype
- Nov 11th: Able to Reload Personal Sticker Map in New Session
- Dec 4th: Design Document Rough Draft Submitted
- Dec 26th: Slideshow Completed
- Jan 2nd: Presentation Recorded for Design Reviews

9.3 Winter 2021

- Weeks 1-2: Design Reviews
- Week 3: Added Authentication to App; Connected App to Firebase
- Week 4: World Maps Save to Firebase; Autosave uploads to Firebase

- Week 5: App Can Load World Map and Associated Stickers
- Week 6: Snapshots for relocalization implemented. Location Manager collects User's Longitude and Latitude.
- Week 7: Unit Testing begins. Added support for 2D stickers in Template Customization.
- Week 8: Added followers and follow requests. Added a feed to discover stickers.
- Week 9: Implemented caching. Added tags to stickers.
- Week 10: Onboarding view finished. Added private account options.

9.4 Spring 2021

- Week 1: App splash screen added. Private sessions implemented.
- Week 2: In-app customer support form added. Users can now invite specific people to their private maps.
- Week 3: Added new 3D models to the app
- Week 4: App published to Testflight
- Week 8: Present Design Report

9.5 Considerations for Further Development

When Apple releases Apple Glasses, we plan to add functionality for glasses (most likely based on eye movement or hand gestures rather than screen taps). Additionally, we are looking forward to adding more functionality to our sticker editing feature. We plan to add custom text size, fonts, and colors, sticker interaction via hyperlinks, and even custom drawings using PencilKit. Finally, we are always taking in feedback via our TestFlight and implementing fixes.

Chapter 10

Risks

Table 10.1: Project Risks

Risk	Likelihood	Predicted Severity	Notes/Mitigation
Lack of Swift Knowledge	Guaranteed	Minor	We are familiar with other programming languages which share the core logic. Documentation and syntax tips are readily available online.
Time Management	High	Variable	Inevitably we are going to run into conflicts for our time between classes, work, and the project. How well we are able to handle this conflict will determine how severely our work is impacted.
Illness	High	Minor	It is not only Flu Season, but also Covid Season, so the chances of one of us becoming ill are high. The impact of illness often varies, but it can take someone away from work for days at a time.
Scalability	High	High	We have concerns about the scalability of the ARWorldMaps that we base our application's understanding of the physical world upon. If we are unable to properly scale them (i.e. splitting the world into map regions and being able to merge map data without conflicts), this would severely impact the performance of our platform. We would have to scale back the application and focus on use-cases that can be enabled by Private Sessions or local peer-to-peer sharing.
Accuracy	Guaranteed	Minor	ARKit is a mature enough framework that most existing mobile Apple devices are built to support Augmented Reality in some capacity. In our testing so far, we have found that while most devices handle horizontal surfaces almost perfectly, they have trouble quickly and accurately detecting vertical surfaces like walls. This vulnerability makes it hard to place stickers on walls, which is a core functionality of StickARs. However, some newer devices like the iPad Pro and iPhone 12 Pro have LiDAR sensors that make it easy to detect vertical surfaces. We are in the process of finding a way to increase the sensitivity of older devices, but if proven impossible, we would have to limit StickARs to iDevices equipped with LiDAR.

Risk	**Likelihood**	**Predicted Severity**	**Notes/Mitigation**
Bugs and Errors	Guaranteed	Variable	Bugs are an inevitability in software development. The severity will depend on how effectively we catch and fix bugs as they crop up. We believe that our test plan should keep this from being too much of an issue. Between unit tests and user testing we hope to catch the majority of bugs - certainly the show-stoppers.
Team Communication	Moderate	Variable	We are collaborating online through Clubhouse to help timeline and divide tasks between team members. Additionally, we communicate through a group chat and weekly meetings to maintain team interaction.
Feasibility	Minor	High	We will need to test that our project meets the standards of our team and the customer base.
Security	Minor	High	There is consideration for the privacy of our users, and our project will need to protect sensitive information. In order for users to make an account, they will have to give us some personal information.

Chapter 11

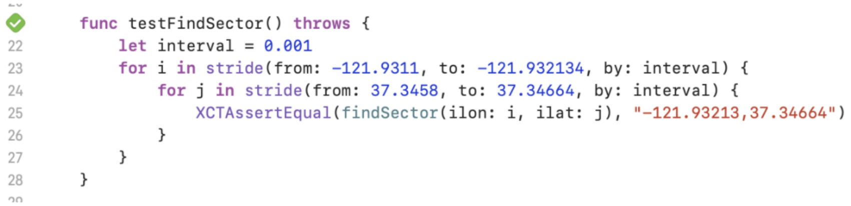
Test Plan & Results

This chapter discusses the techniques we employed while testing our application. Specifically, since StickARs relies on GPS and AR tracking, we had to focus on testing how accurately we could relocalize to a known location. Additionally, we had to test the scalability of our platform to see whether it can succeed as a social networking platform as well. Finally, to ensure the usability of our application and see how potential users would respond to the platform, we used Testflight to hold alpha and beta tests.

11.1 Unit Testing

Xcode provides built-in tools for writing and executing unit tests on your applications. We planned to employ these tools where appropriate to develop unit tests for modular components of our application. For example, we might have a unit test which attempts to download a map from the server and asserts that it succeeds or failure is handled gracefully. Unit tests are a powerful tool for testing modular components of a system, and make it quicker and easier to check for regressions after making changes to the codebase. (Regressions meaning behaviors that previously worked being broken when working on new features).

This is one area of development/testing where we could have done better - there are many modules of our code which were candidates for unit testing but we didn't write tests for them - in some cases, this came back to haunt us when regressions were introduced in these modules. For the modules where we did write tests, it was easy to verify their integrity after making changes, which was useful in speeding up our development.



```
func testFindSector() throws {
    let interval = 0.001
    for i in stride(from: -121.9311, to: -121.932134, by: interval) {
        for j in stride(from: 37.3458, to: 37.34664, by: interval) {
            XCTAssertEqual(findSector(ilon: i, ilat: j), "-121.93213,37.34664")
        }
    }
}
```

Figure 11.1: Unit Test Example
A Unit Test passing in Xcode, verifying that our findSector function still works after changes have been made to the codebase.

11.2 Persistence Accuracy

A large area of concern for us is the precision with which we can re-localize stickers. This is a test which can't easily be automated like our unit tests, because the device needs to be moved around the environment to re-localize to it. As such, our test plan for this feature consisted of saving maps with several virtual sticky-notes placed at the corners of surfaces such as tables, desks, and walls. Upon re-localization, we could determine how far from these corners the stickers have drifted.

In indoor situations with good lighting, and using an iDevice with 2 or more cameras, drift was negligible - generally under 1.5cm. In outdoor situations, the accuracy tended to depend on the quality of the lighting - specifically, how closely the lighting matched from when the space was originally mapped. In worst-case situations, the device would be unable to recognize the map. Typically outdoor situations averaged drift of up to 3cm.

11.3 Scalability

Another important area of concern was how well our platform would scale. Ideally, it should support any number of users around the globe actively placing stickers and loading others'. We planned to test this by slowly ramping up the number of users in our testing. Initially, we tested our backend with just the four of us using the application. Once this was working satisfactorily, we invited some of our friends and family to help us test the application. We also recruited testers from the university community to help beta test the application as it became feature-complete. See 11.4 for more details about this.

11.4 Alpha/Beta Testing with Testflight

The final stage of our testing plan involved utilizing Apple's TestFlight platform to release our application to testers. TestFlight offers a less-restrictive review process than publishing to the App Store, and makes it easy for us to push builds and for testers to download them to their devices. TestFlight also makes it easy for testers to send feedback - if a user takes a screenshot in our application, one of the options in the share sheet will be to send us feedback.

Unfortunately we had trouble attracting testers for the application within our timeframe. We got 7 sign-ups, but only 2 testers installed the application. TestFlight analytics showed that the testers launched the application 14 times and encountered two crashes due to an out-of-memory issue. This was an issue we were aware existed, but it was helpful to see how often it occurred in the field, particularly on their different iPhone models. Had we gotten more testers to install and actively use the application, we would have sent out surveys to our testers to collect more data.

Chapter 12

Societal Components

Our project involves directly interfacing with user data, as well as providing users' with tools to share their speech with others. This introduces a variety of concerns, ranging from ethical issues to protecting the safety of our users. In this chapter, we explain some of the potential issues we've considered, as well as what we've done to ensure they don't overshadow our goals.

12.1 Ethical Concerns

One major ethical concern we have is that we plan to operate a platform that requires the user to map their physical space and upload that map's metadata to our servers. For example, if a user were to place a sticker in their bedroom, a map of their bedroom (stored as a set of points that are used to relocalize) would be uploaded along with the sticker. This is not something that we can avoid - without the map of the physical space, there would be no way to relocalize objects. The data we collect on users is encrypted in our database, so we hope that our users place their trust in us that we are taking the necessary precautions to protect all parties involved. Because we know that various users will likely want different levels of privacy, we implemented private sessions so that users can opt-out of other users potentially seeing their stickers and navigating to them. We also allow users to make their accounts private which prevents other users from following them.

To ensure that no user is getting access to a sticker or account that they shouldn't have access to, we have rules in our code as well as the database mandating which users can see which stickers, users, and WorldMaps. Putting rules on the database prevents unauthorized reads/writes to the database, and putting rules in the code ensures a user does not encounter errors by trying to access information they shouldn't have access to.

Our platform provides the tools necessary to keep our users safe from other users. Users can't harass others with follow requests because any user can disable follow requests in the settings menu. This essentially means users can forego the social feature in our app and use our app exclusively as a private user with private maps. This is how we know we can keep our users safe because they can cut off all contact with other users and use our application for your own private purposes.

Additionally, our application will archive all uploaded stickers for a certain period of time before permanently deleting them. This is to reserve the data in case any sort of unwelcome content is uploaded to our platform, and further consequences can be properly enforced with the evidence archived.

12.2 Sustainability and Environmental Impact

Visual pollution is controlled by the government through the limit of billboards along the highway. In high-population areas like New York, there is no more space for yet another physical billboard. Hawaii, along with some other states, does not allow billboards because of the visual pollution it creates. By converting physical billboards to virtual billboards in AR, it not only reduces visual pollution and preserves the scenic views (thus places like Hawaii would allow it), but it also reduces the pollution created while making the billboard materials. Similarly, virtual stickers are sustainable because they do not need to be repaired. For example, some hiking trail signs get torn down or worn down over time. For somebody unfamiliar with the area, this could create confusion and somebody could get lost. With StickARs, however, virtual trail markers could exist forever (or until the park ranger authorizes them to be removed or changed).

12.3 Social Context

StickArs will provide a hub for communities to form under the basis of creating, using, and/or following similar stickers and/or people. Tags will sort stickers to their respective groups, thus users can find people interested in the same tags as them. For example, if I like mushroom hunting, I might follow #mushrooms and #mushroomhunting. By following these tags I will see any public stickers posted using those tags, thus allowing me to discover new users that I may want to befriend and go mushroom hunting with.

12.4 Political

The nature of this project is not political, however, it could provide a platform for political voices. For example, politicians could place virtual election signs in StickARs for the public to see. These would have several benefits over a standard sign: they can’t be stolen or vandalized, they’re more sustainable because they don’t require any physical materials for something that inevitably gets thrown out when election season ends, and thanks to our preset templates and approval process they are less likely to be an eyesore. The main difference between these and your typical political lawn sign would be that users would not see them unless they follow the user who placed it or a tag that the sticker is marked with. One concern we had was that this might allow users to place themselves in an “echo-chamber” of sorts - only following and seeing stickers from users who share and reinforce their existing views. Unfortunately, this is a concern with most social media platforms, even those which use an algorithm to filter user content because the algorithms are typically personalized. As part of our mitigation for potential abuse, we explicitly designed StickARs to be an opt-in experience - users shouldn't have to fear that they will see something inappropriate or upsetting to them when they load the application in a public space. We believe that this concern and solution does more to serve our users than attempting to prevent them from living in a bubble if they so choose.

12.5 Economic

This app does have a somewhat-high entry cost due to the technology we chose to use. To lessen the entry cost, we broadened the list of supported devices to include non-Lidar devices as well. Lidar improves the functionality and accuracy of our app, but it is not a requirement. That being said, to buy a device that supports this app our users will need to spend at least $200. If we eventually build out the app to be used on Apple Glasses, the entry cost on that platform will be much higher due to the projected cost of Apple Glasses being anywhere at or above $750.

Additionally, using virtual stickers is extremely cost effective when compared to some of the things it can replace. For example, trail markers need constant upkeep and/or replacement, but our stickers stay there until the creator deletes them. Park rangers could place trail marker stickers for a far lower price. Similarly, physical billboards require a huge quantity of material that is not cheap to install or upkeep, but virtual billboards only cost as much as they pay the designer to make the 3D sticker. Even small costs like a daily post-it reminder can be lessened by using virtual stickers instead. In our database we only pay for the amount of data being stored, uploaded, and downloaded, which enables us to offer virtual stickers at a far lower cost than their physical counterparts.

12.6 Health and Safety

As a virtual platform, our users are at a relatively low risk. The most alarming threat to our users' physical safety is the potential of them being distracted while using the app and wandering into a road. Similarly, we worry that a user might try to use the app while driving or biking, thus creating a danger to others on the road. Our team has watched these issues play out with the Pokémon GO AR phenomenon in recent years, with concerning consequences (CNN). To mitigate these issues, we plan on doing two things: first, we will display warnings reminding users to pay attention to their surroundings while using the app; second, we will disable sticker loading while the user is moving at or above a certain speed. Some apps allow the user to manually disable features like this if they confirm that they are only a passenger in the vehicle; considering the potential for abuse, as well as the limited appeal of using our platform while moving at high speeds anyways, we will not offer such an option. Most importantly, however, our app differs from others like Pokémon GO in that it displays an active camera feed, essentially letting the user see through their device.

To aid users' virtual safety and comfort, we allow users to create private maps for stickers they would like to keep to themselves or share with only a select few. This allows users to use stickers to write anything they want, from reminders to passwords. We also implemented private accounts, which turns off a user's ability to be followed. This is useful for a user who is tired of users following him/her, so they can stop being bothered by follower notifications. This also enables a user to be a viewer on public maps while keeping all their stickers private to themselves if they so choose.

Finally, we don't allow users to directly message each other over our platform, so we do not need to worry about censoring chat or storing user's chat metadata. Users are also encouraged to report all offensive stickers to StickARs to maintain a fun, friendly environment. Further regulation of inappropriate stickers could be managed with special tags for age-restricted or potentially graphic stickers.

12.7 Manufacturability

There is no manufacturing involved in this project. Publishing the project will be as simple as pushing a button to upload it to the Apple App Store. Once Apple approves the app for distribution, users will be able to download the app with a push of a button, as long as they have a compatible device.

12.8 Usability

To ensure usability of our app, we have added instructions on most of the views. We also keep the app as minimal as possible, to ensure users don't feel overwhelmed or lost within the app. We did our best to follow convention while organizing the content of the application. For example, the main tab, the AR experience, sits in the center of the app's navigation bar, and the settings can be found on the user's page. We will need to conduct further usability testing to refine the app more.

Our app does not currently accommodate to blind users because it requires looking at the camera to use the app. In the future we could add vibrations when a new sticker comes into view and an audio description to help tell the user what they found. Users with other disabilities such as deafness or mobility impairments should be able to use the app as normal. We currently do not have audio within the app, so as long as the user can see the app, they will get the full experience. Users with mobility impairments can also start private maps within their household that do not require them to travel long distances on foot.

Our app is built with support for localization of all user-facing text. Currently we have only localized the app into English but the groundwork has been laid so that if we had translators for our messages we could easily make the app available in any language. Supporting this feature allows our app to be used in the user's native language which often reduces the usability friction for them. It also allows users who may not know English to make use of our application.

12.9 Lifelong Learning

We hope this application will one day be a hub of creativity that will become another medium for digital artists to display their work. We also hope to offer our app as a framework for other businesses to build off of – for example, a golf course could sell a package that gives golfers access to a StickARs map that marks where and how hard to hit the balls for the best shot. Similarly, a national park could sell a package that gives access to a StickARs map that has trail markers placed as well as other indicators for falling rocks, edible mushrooms, and dangerous animal sightings.

We also set up our code such that future versions of the app will not break past versions. That means if user A is using a new version with more stickers and user B is on the older version without those stickers, then user B will be able to use the app as normal, but they may not be able to view stickers placed by user A that are not in user B's version of the app. We hope that this will help us make continuous improvements to the app without the worry of breaking the app for users who don't update right away.

As the platform grows, we may need to improve upon the regulation of offensive content within the app. As of now, the app has preset stickers so the biggest worry is that users will write offensive text on the stickers. In the future we want to allow users to submit their own stickers for approval, and this will require much more work on our part. We will have to make sure the sticker cannot be used in a blatantly offensive manner.

12.10 Compassion

This project is inspired by our society's increasing demand for VR and AR products. While Apple has yet to release their Apple Glasses, we hope that by developing this app now, our AR technology can improve by the time our framework is compatible with Apple Glasses.

The virtual nature of this project does not directly relieve the physical suffering in communities, however, the compassion of our project is delivered through forming communities with others who have similar interests. StickARs is meant to be a source of fun, relaxation, and inspiration in lieu of COVID-19 limiting socializing.

Chapter 13

Conclusion

13.1 Lessons Learned

StickARs grew into a much more complex project than we initially expected because we saw a great deal of value in the service it performs: bringing people together through the connection of the virtual and the real worlds. Some of our team members had never developed an app prior to StickARs, and most of us didn't know how to code in Swift at all, yet the project motivated us to embrace this new challenge.

Building StickARs was not easy. Not only was the project itself technically challenging, but the COVID-19 pandemic also presented us with a new set of challenges: working as a remote team in different time zones.

On the technical front, we set out to build a world-scale Augmented Reality experience that runs seamlessly while the user navigates the world. We fell into the trap of picking our desired hardware at the beginning of our project, which ended up limiting us to using ARKit as our AR API. This limitation meant that we could only harness as much world data as is available to us from the API. The API is not built for world-scale experiences, so we had to be creative in our development of StickARs, which led us to use WorldMaps to enable the user's traversal through the real world.

Logistically, this project was troublesome, because even though it didn't require any hands-on development, it was still difficult to coordinate with four team members scattered across the U.S. Early on, we had to learn how to use project management methodologies and software like Trello and Clubhouse to facilitate our development. Learning how to properly use source control software like Git was also crucial to our success. Finally, navigating communication between team members was also a challenge, as this was the first time many of us worked on a long term project with a team.

13.2 Advice for future development

Similar projects should not be taken lightly. We believe that to achieve an optimal result, we would need to build our own AR API bespokely designed to facilitate world-scale AR tracking, a skill we don't currently possess. Current AR APIs all focus on small-scale experiences, which makes it difficult to develop compelling AR software. These APIs also have a tougher time with outdoor environments, as lighting conditions can vary from blindingly bright to pitch black, and the user's device will have trouble identifying its location relative to the world, something that needs to be accounted for in future development.

On the non-technical side, we learned that open communication is key to the success of the team, and by extent, the project itself. We also saw the importance of scheduling work appropriately in order to make enough progress to stay on track but not overwork ourselves in the process, and gave ourselves as much help as possible by using the best available tools to facilitate development GitHub and Clubhouse.

Had we limited our ambitions of creating a world-scale AR experience, we may have likely had a much more polished product at this point. However, our ambition allowed us to really push ourselves to learn as much as we could in the given time frame, and creatively push the boundaries of what our chosen technology is capable of.

Bibliography

[1] Apple Inc., “Swift.org.”

[2] Apple Inc., “ARKit | Apple Developer Documentation.”

[3] Apple Inc., “SceneKit | Apple Developer Documentation.”

[4] Apple Inc., “SwiftUI | Apple Developer Documentation.”

[5] Google Inc., “Firebase API Reference.”