# Project Status Sheets

# Preliminary Project Proposal

Parkit is your go to app when it comes to parking. The aim of Parkit is to ease the pain of parking by allowing homeowners to rent out their driveways for an allotted amount of time. It is also a great source to make passive income. The owner can make money by renting out their driveway while they are at work or on vacation. Instead of paying $20 flat rate you can pay half that while the homeowner makes a nice sum.

# Proposal Document and Presentation Slides

# Verbal Description

# Parkit is a mobile application similar to Airbnb. It allows a homeowner to rent out their driveway for others to park in. It serves two groups of users classified as Parker and Owner. The Owner is the user who owns the driveway. He or she will be able to register their stall (driveway) via the application and set the hours in which their stall will be available to park. The Owner has the ability to transfer their collected funds and check the status of their stall. The Parker will have the ability to reserve, check-in and check-out a stall.

# Justification

# This project will involve knowledge learned from various classes offered at LMU, like Computer Programming, Data Structures, Algorithms, Interaction Design, Programming Languages and Databases. While creating a mobile application does take a lot of time and effort it should not take me more than a whole semester when shooting for a minimal viable product. And while this specific idea has not been implemented by previous LMU students, there have been numerous amount of LMU students have chosen to develop a mobile application for the class.

# (See PARKIT presentation slide)

# Software Development Plan

* 1. This Software Development Plan provides the details of the planned development for the PARKIT Application CSCI. PARKIT is a mobile and web application that allows users to rent out their driveways and allows for users to park in available driveways. The aim of Parkit is to ease the pain of parking by allowing homeowners to rent out their driveways for an allotted amount of time. It is also a great source to make passive income. The owner can make money by renting out their driveway while they are at work or on vacation. Instead of paying $20 flat rate you can pay half that while the homeowner makes a nice sum.
     1. Project Deliverables
        1. February 10th - User reporting is implemented and all high priority features are implemented
        2. March 11th - Admin GUI is implemented and Medium priority features should be implemented
        3. April 3rd – Application enters alpha
        4. May 1st- Application is scheduled for Beta
        5. June 5th- Parkit 1.0 is scheduled for release
  2. Project Resources

# Requirements Document

## Introduction

PARKIT is a mobile and web application that allows users to rent out their driveways and allows for users to park in available driveways. The remainder of this section is structured as follows. Section 5.2 contains functional requirements, section 5.3 contains performance requirements and section 5.4 contains environment requirements. These requirements are expected of the initial full-release of the iOS app scheduled to be released in June.

## Functional Requirements

### General System Requirements

* + - 1. The system shall allow the user to register a parker account
      2. The system shall allow the user to login to their parker account
      3. The system shall display messages to the user designated from an admin
      4. The system shall allow the user to fill out a form to notify an admin of any issue such as a bug, inappropriate content etc.
      5. The system shall allow the user to link a credit or debit card to their account.
      6. The system shall allow the user to modify their email
      7. Upon reserving a stall the system should provide directions to the stall that the user reserved.
    1. General GUI Requirements
       1. The GUI shall display a splash screen when the application is loading
       2. The GUI should display a loading animation when the application is conducting a http request.
       3. Stall Searching Requirements
          1. The GUI shall display an interactive map of your current location
          2. The GUI shall allow the user to type an address into a search bar.
          3. The GUI shall display the closest stall available within a mile radius of the address inputted by the user.
          4. If there are no stalls available but are there closed or reserved stall within the radius the GUI shall display that location
          5. If there are no stalls within the address entered the GUI will display a message stating, there are no stalls within the area.
          6. The GUI shall display the hourly rate of the stall
       4. When a stall is found and available the GUI shall allow the user to mark the spot as reserve for a maximum allotted time of 1 hr.
       5. The GUI shall display an option to change the form of payment.
          1. The system shall allow the user to pay via the user’s debit or credit card
          2. The system shall allow the user to pay via Apple Pay
       6. The GUI shall allow the user to check-in to the stall via a button displaying the message ‘PARK’
       7. The GUI shall display Parking Instructions provided by the stall owner.
       8. The GUI shall display the check-out time information provided by the stall owner.
       9. The GUI shall allow the user to check out of the stall via a button displaying the message ‘CHECK OUT’
          1. Upon check out the GUI shall display a receipt

The receipt shall display the length of stay.

The receipt shall display how much the user has been billed

The receipt shall display the address of the stall

The receipt shall display the time of check-in

The receipt shall display the time of check-out

* + - * 1. The user shall be given the option to confirm the receipt via a button displaying the message ‘Confirm’
        2. The user should be given the option of clicking a button displaying the message ‘Need Help’
        3. The user should be given the option to rate the user via clickable ‘stars’
    1. Stall Owner System Requirements
       1. The system shall allow the user to register a stall location
       2. They system shall allow the user to enter their address
          1. The system should verify that the address is correct
       3. The system shall allow the user to login to their home owner account
       4. The system shall allow the user to change the times their stall is available.
       5. The system shall allow the user to see the status of their stall, whether if it is available, reserved, in-use or closed.
       6. The system should allow the user to add a photo of the parking location to their house
       7. The system shall allow the user to add special parking instructions for their stall.
       8. The system should allow the user to transfer their account balance to their bank account.
    2. Parker Owner System Requirements
       1. The system shall allow the user to add their car information to their account.
       2. The system shall allow the user to attach a photo of their car to their account.
       3. The system should notify the user if they’re close to the check-out time.
       4. The system should notify the user if their vehicle is being towed.
    3. Admin System Requirements
       1. The system shall allow the admin to login via an account created and provided by the PARKIT technical officer or database administrator.
       2. The system shall allow the admin to modify parking rates of stalls.
       3. The system shall allow the admin to modify parker and stall owner accounts.
       4. The system should allow the admin to send messages to a single user or all users within a specific group.
       5. The system shall allow the admin to view error reports
  1. Performance Requirements
     1. The system shall support up to 20,000 users.
     2. All searches conducted within the system shall only take up to 15 seconds.
     3. Servers shall be online 120 hours out of 168 hours a week.
     4. A registration request of any account shall only take up to 10 seconds.
     5. The application shall require no more than 15% CPU utilization
     6. The database shall be able to handle at least 1,000 read transactions per hour.
  2. Environment Requirements
     1. Development Environment Requirements
        1. The iOS application will require xCode and swift 3
        2. The server shall require nodejs to develop
        3. Versioning Control will be handled by git and the repository will be hosted on github
        4. The database will be written in MySQL
     2. Execution Requirements
        1. Hardware Requirements
           1. Mobile

The minimum requirement device to run the application on iOS is the iPhone 4.

* + - * 1. Web Application

Any device that supports the browser software requirements

* + - 1. Software Requirements
         1. The iOS mobile application shall be accessed through devices that use iOS 9.0
         2. The web application is officially supported by the following browsers. Google Chrome 53.0+[[1]](#footnote-1), Firefox 50.0 + and Safari 9.1+

1. Indicates support for current and future releases. [↑](#footnote-ref-1)