Report # 1

Course Title: Software Engineering Spring 2019

Group Number: 19

Project Title: Rutgers Parking System

URL of Project Website:

https://sites.google.com/a/scarletmail.rutgers.edu/parking-garage-se/

Submission Date: February 24, 2019

Team Members:

Suva Shahria, Krithika Uthaman, Andrew Schneeloch, Josh LoGiudice, Gabriel Shen, Anthony Lau, Jahidul Islam, Yu Liu & Max Davatelis

Individual Contributions Breakdown

	Suva Shahria	Jahidul Islam	Krithika Uthaman	Josh LoGiudice	Gabriel Shen	Yu Liu	Andrew Schneeloch	Anthony Lau	Max Davatelis
Project Management (10 points)	60		40						
Sec.1: Customer Problem Statement (*9 points*)	30	30	30			10			
Sec.2: System Requirements (*6 points*)	25	25	25				25		
Sec.3: Functional Requirements Specification (*30 points*)	15	20	15	10		15	10	15	
Sec.4: User Interface Specs (*15 points*)				20	20		20		40
Sec.5: Domain Analysis (*25 points*)	12.5	12.5	12.5	12.5		12.5	12.5	12.5	12.5
Sec.6: Plan of Work (*5 points*)					100				

Suva = .6*10+.3*9+.25*6+.15*30+.125*25 = 17.825
Jahidul = .3*9+.25*6+.2*30+.125*25 = 13.325
Krithika = .4*10 + .3*9 +.25*6 +.15*30 + .125*25 = 15.825
Josh = .10*30 + .20*15+.125*25 = 9.125
Gabriel = .2*15+1*5 = 8
Yu = .1*9 + .15*30 +.125*25 = 8.525
Andrew = .25*6+.1*30+.2*15+.125*25 = 10.625
Anthony = .15*30 +.125*25 = 7.625
Max = .4*15 + .125*25 = 9.125

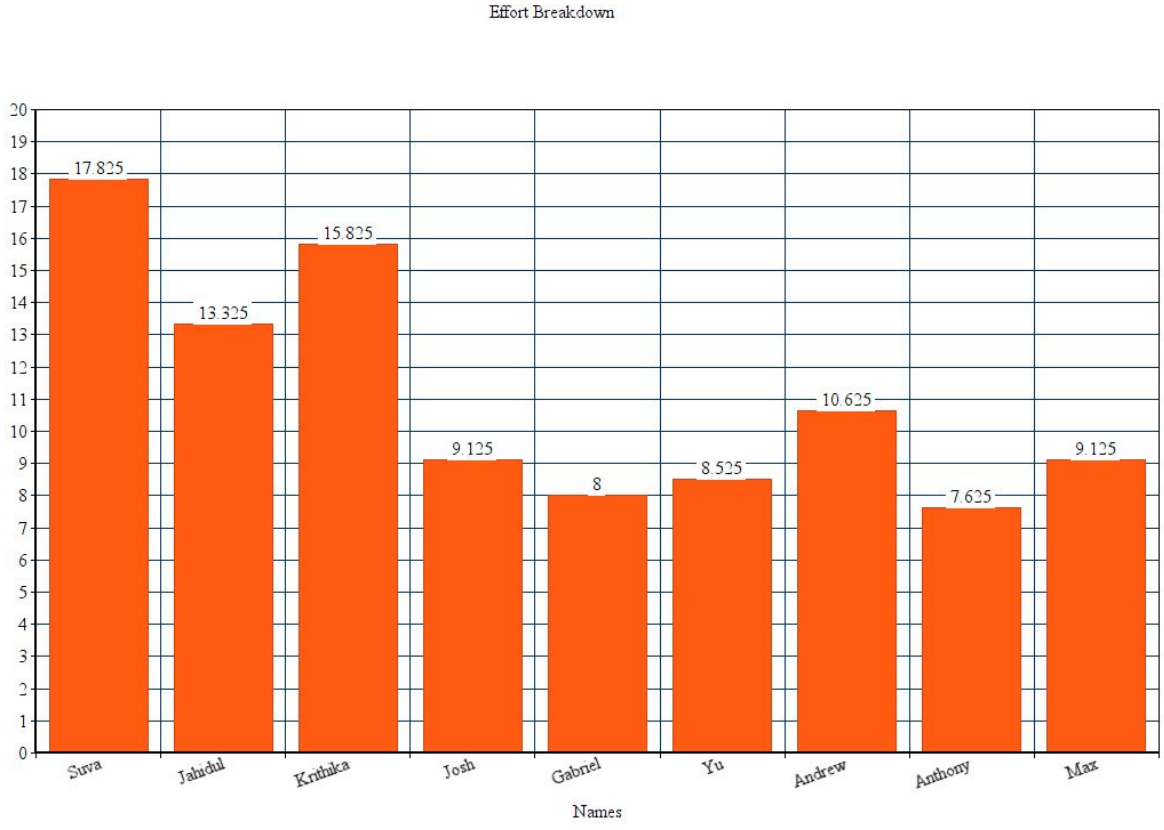


Table of Contents

Section 1: Customer Problem Statement

Problem Statement

With a continuously rising population density, large issues arise with simple accommodations such as parking. Diminishing open parking spaces often are left unresolved even with the addition of new parking garages with their currently inefficient allocation of reserved spots. Certain ventures have been made to automate the process through a website or app, but still leave room for improvement. Many parking spaces still remain with empty unused portion for varying amounts of time with the potential of supplying more consumers with available space. Creating new facilities without improving the currently existing only increases congestion in new areas. In order to fully take advantage of current parking lots and garages a new system is needed. With each parking spot being efficiently utilized, congestion will become highly limited.

In addition to maximizing parking usage, another issue affecting parking decision lies with confusing regulations. With a plethora of parking systems, some with their own specific rules, a customer's confusion leads to incorrect parking which then in turn affects other customers creating greater congestion. Pairing such confusion with an event which requires large parking accommodations leads to visible evidence regarding the faults of the current system. Overall, the new system must recognize available space immediately as they become available in order to ensure proper utilization.

In choosing a system requiring more information to supply users with possibilities, more information in turn is required from the users creating security concerns. The consumers data must be properly secured both on the app and on the system utilized for confirmation of their identity upon arrival at the parking lot or garage. In protecting their information, a peace of mind is gained while data of their usage can be utilized for helping other customers park.

A specific example of these issues with parking are found within the Rutgers University community. Rutgers students, faculty, and staff depend on the parking system to be reliable and efficient especially with a huge portion of the student body being commuters. Instead they are dealing with the confusing system currently in place which is unclear on which lots an individual

is allowed to park in and when, which then makes it difficult for students to get to class on time. Currently a common occurrence is many vehicles frequently and inefficiently roaming around the lots looking for an ideal parking spot and sometimes following students back to their spots. Individuals currently show up to parking lots clueless to the actual availability and resort to the problematic pattern of searching for a spot. Many times the students receive parking tickets despite the fact of parking in the correct lot or due to small technicalities in the parking guidelines.

The parking issues are then amplified when considering special events, such as football games. These events bring massive amounts of people and vehicles who aren't usually here. The inability to predict parking accommodations for such occurrences hinder the situation further leading to prolonged searches and confusion.

The old fashioned method of just creating more space does not tackle the core of the issue. Furthermore, in prioritizing creation of more parking spots it may reduce overall transportation as it neglects space to accommodate other modes of travel such as biking or even possible locations for public transport such as bus stations. By reducing options for other modes of transport, the congestion dilemma remains prevalent. An increase in traffic surrounding the new garage is also to be expected. A consumer focused improvement on existing garages is needed rather than an increase of capacity.

Proposed Solution

In beginning the approach to solve the given issue, first, data is needed from the customer. Specifically, their arrival time, time parked, as well as exit time. This will provide the app's database with required information needed to recommend parking areas with availability. The app utilization of login and identification info such as ID will be linked to their parking tendencies which will be protected via encryption. That data of when the parking spots are vacant or occupied will allow the app to properly dictate to parking seekers exactly where parking is available.

The solution to the parking issues at Rutgers is an app which all individuals of the Rutgers community can download. An app would be more suitable than a website since most of the users demographic are expected to own smartphones and it would be easier to use on the go.

Users will be able to create their own accounts, with a username and password, where they will enter their parking permissions, for example a user may hold a "Commuter Zone A" parking permit. The app will let them know which lots they can park in according to their parking permissions, the occupancy of that lot and the day/time. There will also be an interactive map which will let the user know which spots are vacant and which are occupied.

As the user uses the app more and more often, the app will be able to more efficiently recommend certain spots. The system will also keep track of a user's favorite spots and log how long they usually stay there. The user can input where their classes take place so that the system can recommend spots accordingly.

Although the solution would work whether or not everyone who parks at Rutgers uses the app, the efficiency of the system would increase as more of the community uses it. The system will be able to track the usage of certain lots and track how busy they are over the course of the day as well as throughout the week. By comparing the actual occupancy of the lot to the average occupancy the system will be able to offer better parking suggestions.

For special cases, such as football games, the system will be notified that certain lots, like the ones around the stadium on Busch, are currently off limits and the system will respond accordingly. In order to accommodate the large influx of new parkers, certain areas will be reserved for visitors only as well in order to minimize negative effects for students.

Assumptions

In order for this solution to work, some assumptions need to be made. This includes that all users have smartphones and have access to the app. The second assumption is that there is a spot-occupancy recognition system in place that works correctly 100% of time. This system would keep track of whether spots are currently vacant or occupied. The choice of system will be a scanner capable of scanning a student or or faculty member's Rutgers ID.

Glossary of Terms

- Account: Customer creates an account with a username and password along with their parking permissions which holds all the users information
- App: Mobile application that customers can log into and will help recommend them parking spots
- Database: Store user's account information and parking guidelines that feeds data to the app
- Encryption: The act of encrypting user information so that personal information can't get stolen
- Interactive Map: Maps of every parking lot/garage which shades occupied spots in red and vacant spots in green
- Occupied Spot: A spot which has a vehicle in it
- Parking Permissions: Where a user is permitted to park
- Registration: Creating an account on the app
- User: A person who uses the app and intends to find an ideal parking spot
- Vacant Spot: A spot which doesn't have a vehicle in it

Section 2: System Requirements

Enumerated Functional Requirements

Identifier	Priority	Requirement
REQ-1	5	The system will let users create, edit and delete their own accounts through the app. Account information includes username, password and the user's parking permissions
REQ-2	5	The system will have the current status (vacant, occupied) of every parking lots based on the number of people scanned in
REQ-3	4	The system will have an occupancy chart for every parking lot
REQ-4	5	The system will operate with the notion of statuses of passes for authorization i.e. guest pass, student pass, etc.
REQ-5	4	The system will let the user know which lots they are permitted to park in
REQ-6	2	The system will issue a warning if the lot they have chosen to park in has certain restrictions
REQ-7	3	The system will be aware of any special cases (football games, a lot is being closed off, etc.)
REQ-8	2	The system will keep know the occupancy patterns of lots throughout the day and throughout the week by monitoring scanned in IDs
REQ-9	3	The system will track various stats of lot usage and efficiency by utilizing data such as occupancy through ID scan in

Enumerated Nonfunctional Requirements

Identifier	Priority	Requirement
REQ-10	3	The app will track users and when they park in the lot and compare that to how many spots are in the lot this will allow the app to give a more accurate occupancy chart the more the app is being used
REQ-11	2	The system will have a log of past/favorite lots/spots the user has parked in and for how long
REQ-12	2	System will allow users to view history of parked locations
REQ-13	1	The stats collected by the system can be passed on to the Department of Transportation at Rutgers if they are interested in using it to improve their own system

On-Screen Appearance Requirements

Identifier	Priority	Requirement
REQ-14	5	The app should have an initial page where you sign in, with a link that says if you don't not have a account make one with a link to register for an account
REQ-15	5	The registration link will take you to a page where you will fill out your information including status i.e. student, guest, etc.
REQ-16	4	Once signed in the app will display a menu of campuses that you can park on with the ability to choose one
REQ-17	4	Once a campus is chosen another menu will be displayed with possible parking lot locations on that campus
REQ-18	3	The user will then be allowed to select on a lot and check the status of the lot's occupancy
REQ-19	2	The app will have a profile page where users can edit or delete their accounts
REQ-20	3	The app will have an interactive map for viewing lot areas
REQ-21	3	The app will have a chart showing how busy lots are expected to be throughout the day as well as a comparison of how busy the lot actually is currently

Section 3: Functional Requirement Specification

Stakeholders

- Parkers
    - Students/Professors: Need reliable information on parking availability to attend or teach classes on time
    - Other Users: Seek parking for any variety of reasons and would like to resolve parking issues efficiently
- Department of Transportation: Needs to monitor transportation services and information regarding which parking is used to provide an outlook of how many passengers will be using public transportation and from where. This will allow efficient public transport
- Event Planners: Organizing large events require ample parking in order to manage both attendance and road congestion. By supplying customers with more options, the negative effects can be reduced
- Drivers: Anyone driving near parking facilities can be met with unwanted traffic resulting from people searching for parking as they enter and leave garages. By providing the parker with exact places to park, road congestion will be reduced
- Services
    - Federal/State: Services, like the mail as well as school buses, need safe places to stop and are impacted by traffic and road congestion with people seeking to park or parking incorrectly on the streets lead to negative effects
    - Company: Companies, like Amazon, depend on delivering efficiently and traffic congestion in city residential areas due to parking searches hinder potential profit.
- Creators: The app development team

Actors and Goals

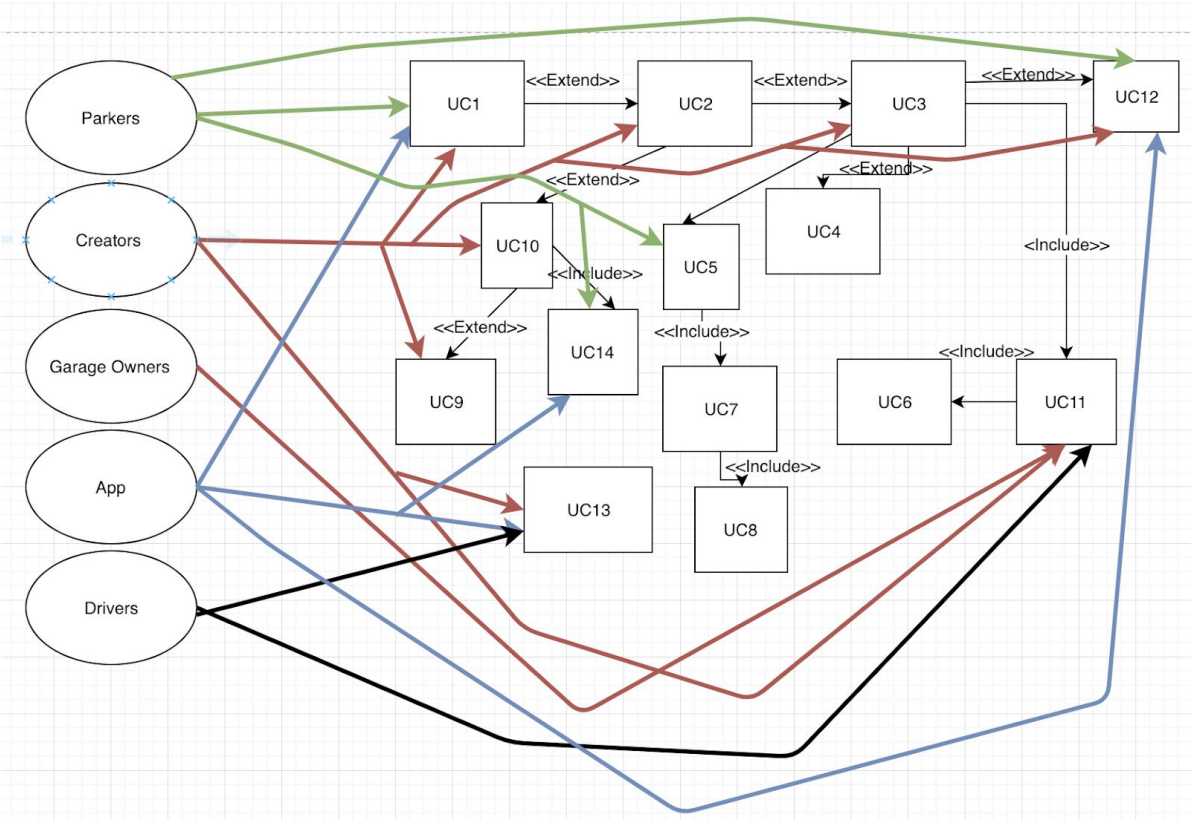
Actors	Roles	Goals
Parkers	Initiating	Find parking efficiently and know the exact destination to avoid hassle
Department of Transportation	Offstage Actor	Utilize info on parking use to efficiently place public transportation
Event Planners	Offstage Actor	Limit congestion during planned events by providing customers with parking solutions
Drivers	Supporting Actor	Alleviate traffic headaches near parking locations by minimizing the people wandering about
Services	Offstage Actor	Less traffic congestion will allow other services to function efficiently
Creators	Supporting Actor	Maintain database and update app
App	Initiating Actor	Used to reserve parking spots as well as indicate which are available. Exiting after parking spot usage is done also provides notification to the system that this spot is ready to be assigned to someone else.

Use Cases

Casual Description

Use Case Identifier	Use Case Name	Description
UC-1	Register	The user will be able to register for a new account within the app
UC-2	Login	The user will be able to login to an existing account from the app
UC-3	Choose Campus	A list of campuses will be listed. The user can choose which campus they want to park on
UC-4	View Spots	A display of the parking lot will be displayed that show both vacant and occupied parking spots
UC-5	Pick Spot	The user can choose a parking spot that is available from the app
UC-6	Events	The system will notify the which lots are closed due to any events
UC-7	Arrival Time	The system will record the time the user arrives in their spot
UC-8	Exit Time	The system will record the time the user leaves
UC-9	Delete Account	The user can delete their account if they want
UC-10	Edit Account	The user can edit any account information such as changing their password
UC-11	Lot Information	The system will display all available parking lots for the user as well as status
UC-12	Interactive Map	The system will display an interactive map of the lots and surrounding area
UC-13	Patterns	The system will keep track of expected occupancy patterns of the lots over time and days of the week

Use Case Diagram



Traceability Matrix

Req.	**Priority**	UC-1	UC-2	UC-3	UC-4	UC-5	UC-6	UC-7	UC-8	UC-9	UC-10	UC-11	UC-12	UC-13
REQ-1	5	X	X							X	X			
REQ-2	5				X							X	X	
REQ-3	4				X							X		X
REQ-4	2			X	X	X								
REQ-5	4		X											
REQ-6	2						X					X		
REQ-7	3						X							
REQ-8	2											X		X
REQ-9	3											X		X
REQ-10	3											X		X
REQ-11	2		X			X		X	X					
REQ-12	2					X								X
REQ-13	2		X											
REQ-14	1													X
Max PW														
Total PW														

Fully-Dressed Description

Use Case: UC-1 Register
Related Requirements:REQ-1, REQ-4, REQ-14, REQ-15
Initiating Actor: Customer
Actor's Goal: To register an account on the App
Participating Actors: System, Customer
Preconditions: The system will ask the customer information to register.
Postconditions: The customer will have created an account that will be store in the database.
Flow of events for main success: - > 1. The customer opens the app to register an account - < 2. The system requests necessary information from the customer - > 3. The customer enters in the necessary information - < 4. The system registers the username into the database for the customer Flow of Events for Extensions (Alternate Scenarios): 2A Username is already taken in the database - < 1. System detects an error - < 2. System notifies customer of error - < 3. System returns user to the registration page

Use Case: UC-5 Pick Spot
Related Requirements:REQ-2, REQ-5, REQ-6,REQ-16,REQ-17, REQ-18
Initiating Actor: Customer
Actor's Goal: Customer can pick an open parking spot from the app
Participating Actors: Customer, System
Preconditions: The system stores information about the current status of all parking slots as open or closed.
Postconditions: The system will update the status of the chosen parking spot
Flow of events for main success: - > 1. Customer will access the app to choose a parking spot - < 2. System will display a visual representation of the parking spots. Each spot will indicate whether they are open or closed

- > 3. The customer will choose an open parking spot to take
- < 4. The system will update the status of the chosen parking spot
- < 5. The System will show a confirmation message

Flow of Events for Extensions (Alternate Scenarios):

2A User chooses a closed spot

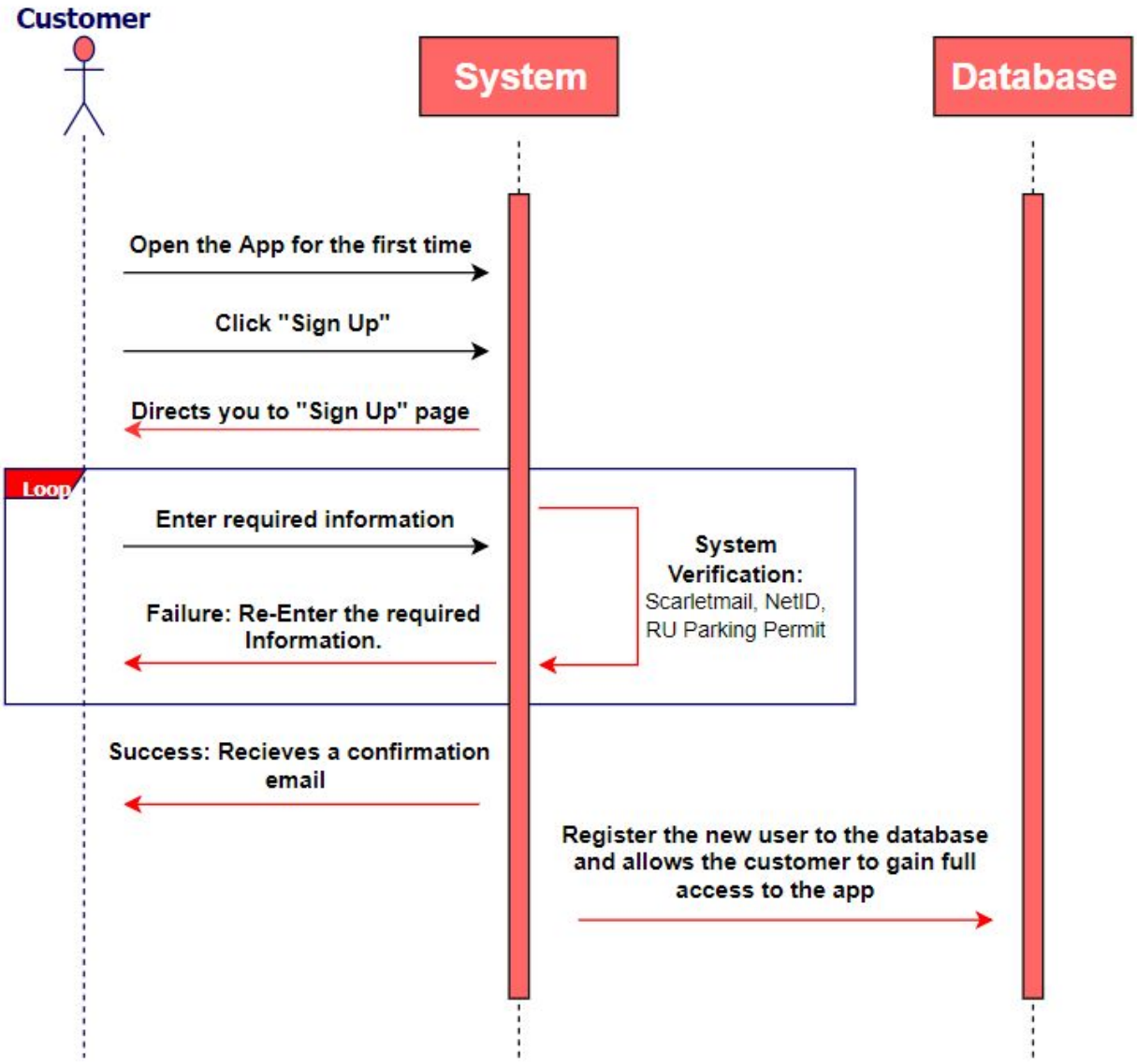
- < 1. The system will give a warning message

3A An open spot is taken as the customer is browsing

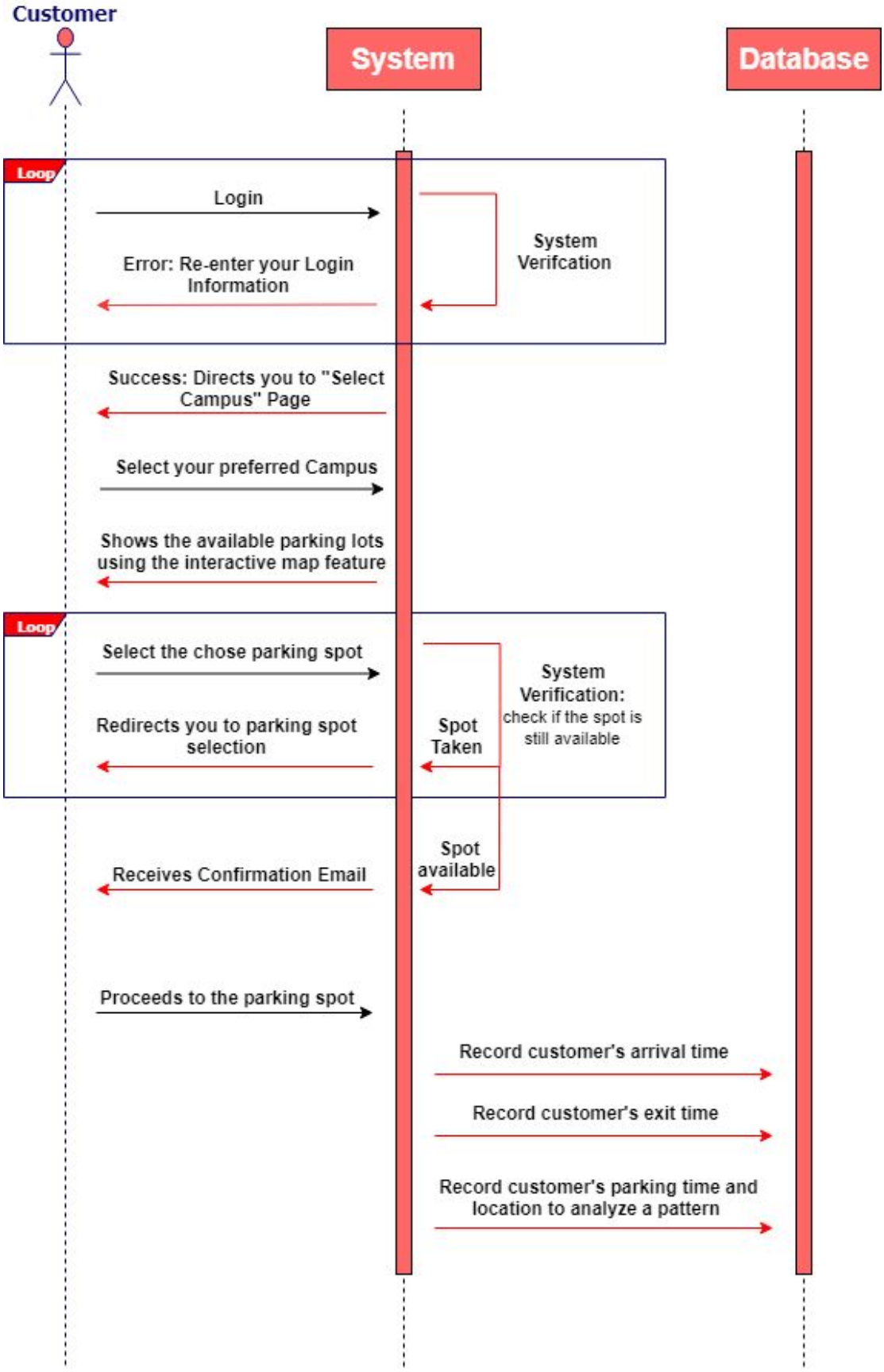
- < 1. The system will update the visual representation of the parking spots to match the current state of the slots

System Sequence Diagrams

UC-1: Register

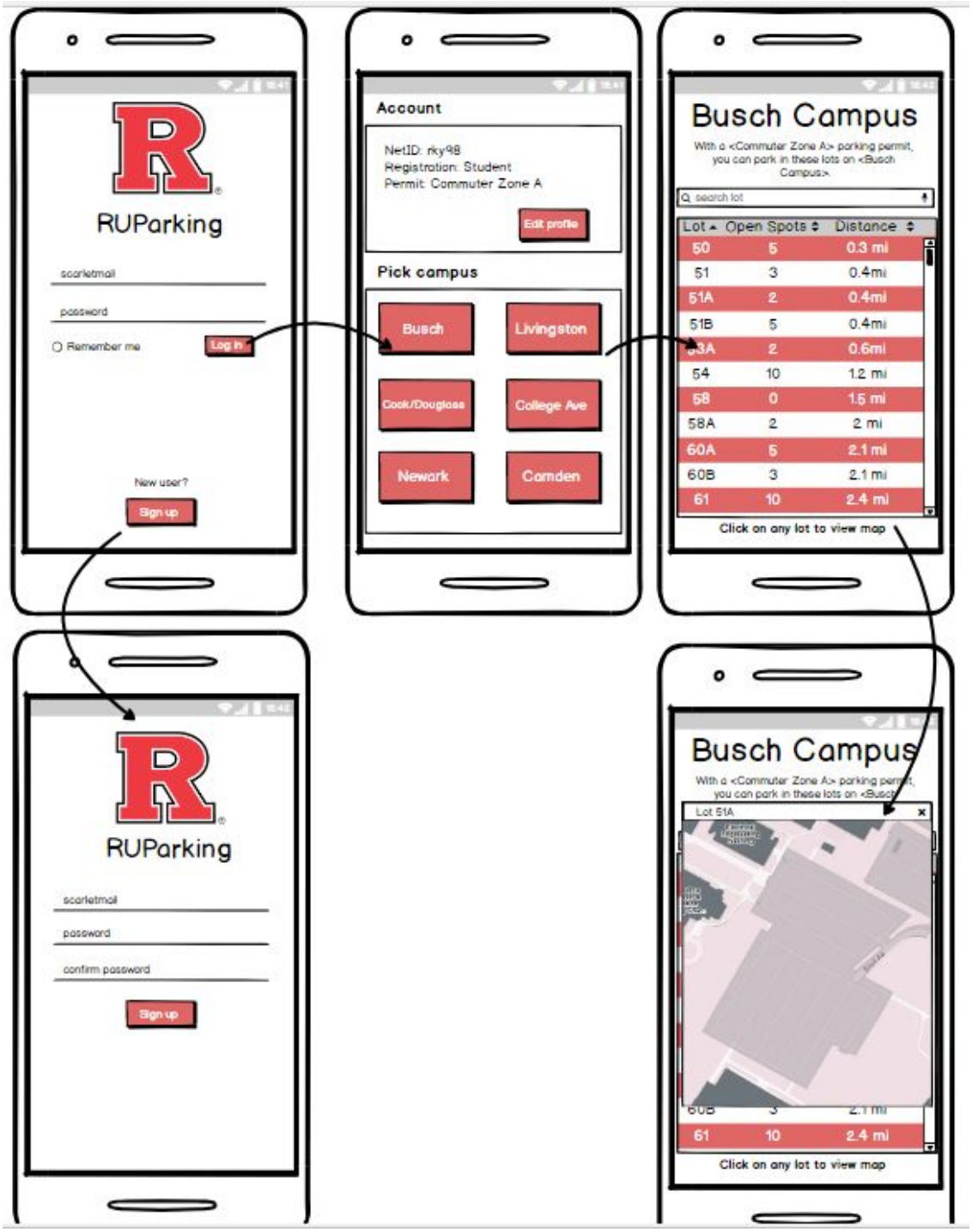


UC-5: Pick Spot



Section 4: User Interface Specification

Preliminary Design



User Effort Estimation

Login:

1. Data Entry: total 1 mouse click and more than 2 keystrokes
    a. Click the “Username” text field
    b. Press the keys for your “Username”
    c. Hit the “tab” key to move to the “Password” text field
    d. Press the keys for your “Password”
    e. Press “Enter” to finish

Getting Parking Spot:

1. Navigation: 3 mouse clicks
    a. Click which campus you would like to park at
    b. Click which lot you would like to park at from available spots
    ---Enter Data---
2. Data Entry: 1 mouse click, 1 text field and a key press
    a. Click the “Parking Time” field
    b. Press the keys to enter the time
    c. Press Enter to finish

Register:

1. Navigation: 1 click
    a. Click “Sign up” from first screen
    ---Enter Data---
2. Data Entry: 1 click, 3 text fields, 3 key presses minimum
    a. Click “scarletmail” text field
    b. Fill in text field
    c. Press the “tab” key to move to the next text field
    d. Fill in the “Password” text field
    e. Press the “tab” key to move to the “Confirm Password” field
    f. Fill in the “Confirm Password” text field
    g. Press the “tab” key to move to the next text field
    h. Fill in the “Status” text field
    i. Press the “Enter” key to finish

Section 5: Domain Analysis

Domain Model

Concept Definitions

UC-1 Register

Responsibility	Type	Concept
Coordinates actions of concepts associated with this use case and delegates the work to other concepts	D	Controller
The visible pages associated with the options to sign in or sign up	K	Interface Page
Prepare a database query with information on existing accounts and credentials	D	Database Connection
The system will make sure no data matches with existing credentials for login are made	D	Postprocessor
Initializes sign up process started by user	K	Sign Up Request
Add sign up information to database for new user	D	Archive
Notify user of successful account registration	D	Notifier

UC-5 Pick Spot

Responsibility	Type	Concept
Coordinates actions of concepts associated with this use case and delegates the work to other concepts		Controller
User specifies desire to search for available parking spot		Parking Spot Request
Database records of nearby available parking spots as well as occupancy charts will be requested		Database Connection

Shows the current context of available spots, and what actions can be taken for desire outcome		Interface Page
Retrieved data must sorted by available and unavailable parking spots		Postprocessor
Retrieved data converted to map display for user		Page Maker
Updates database based on selected parking spot and amount of time desired		Updater
Archives parking request under account history		Archiver
Notifies user of selected parking spot if valid		Notifier

Association Definitions

Concept Pairs	**Association Description**	**Association name**
Interface Page←→Controller	The user can choose to login in to an existing account, or create a new one.	Conveys request
Controller ←→Sign Up Request	The controller sends a request to create a new account, with information provided by the user.	Conveys request
Sign Up Request←→Archiver	The user enters their information for a new account.	generates
Archiver ←→ Database	The new account information received by the database is stored.	Provides data
Interface page ←→ Parking Spot Request	The user will select where he/she wants to park based on the lot number and the available parking spots.	Conveys request
Database←→Postprocessor	Convert Database Info Into User Display.	Provides data
Postprocessor←→Interface Page	Display Choices Based On User Request.	Provides data

Updater←→Database	Update Database Based on User Selection.	Provides data
Archiver←→Database	Archive requests to save user data in the database.	Request save
Notifier←→Interface Page	Notify User of Selections that are invalid.	Request notify
Page Maker ←→ Database	The database passes the retrieved information to the page maker to be displayed.	Provides data
Page Maker ←→ Interface Page	The page maker prepares the display for the interface page.	Prepares

Attribute Definitions

Concept	**Attributes**	**Attribute Description**
Interface Page	Register	Creates a new customer account by requesting the customer's information.
	Login	The App will ask the customer's user ID and password to login.
	ManageAccount	The customer is able to edit and update the user info or delete the current profile.
	SelectCampus	Customer chooses which campus to park in.
	SelectLot	Customer selects the parking spot.
	CancelLot	Customer is able to cancel the current parking lot so he can choose another.

Database	UserInfo	The App will contain user's info such as Scarletmail, NetID, RU Parking Permit, Driver's License, phone number, etc.
	UserID	Customer's username for the app
	UserPassword	Customer's password to login.
	RecordTime	Records the time to the database when the customer enters and exits the parking lot.
	DataAnalysis	The systems records the customer's parking patterns to predict and locate the best parking lot in the future.
	UserStatus	App takes into consideration user's status to determine which lots are permitted
Page Maker	DisplayMap	The system shows the map of the campuses and its associated parking lots.
	LotStatus	Shows which parking lots are currently open and which are closed.
	LotNumber	The system displays the selected parking lot number in which the customer will proceed to park.
Notifier	UserEntrance	Notifies the database when the customer has entered the parking lot.
	UserExit	Notifies the database when the customer leaves the parking lot.
	NotifyUser	The system sends helpful notifications to the customer such as: confirmation emails, how much time it is left, any special events, which campus has more open parking lots, etc.

Traceability Matrix

Use Cases	**Domain Concepts**							
	Interface Page	Database	Page Maker	Notifier	Archiver	Controller	Updater	Post Processor
UC-1	**X**	**X**		**X**	**X**	**X**		
UC-2	**X**	**X**				**X**		**X**
UC-3	**X**		**X**					
UC-4	**X**		**X**					**X**
UC-5	**X**	**X**	**X**	**X**	**X**	**X**	**X**	**X**
UC-6				**X**			**X**	
UC-7		**X**		**X**	**X**		**X**	
UC-8		**X**		**X**	**X**		**X**	
UC-9	**X**	**X**			**X**	**X**		
UC-10	**X**	**X**			**X**	**X**	**X**	
UC-11			**X**				**X**	**X**
UC-12			**X**			**X**	**X**	**X**
UC-13		**X**			**X**		**X**	

System Operation Contracts

Operation	Register
Preconditions	Customer has not registered into the system Customer will enter valid information
Postconditions	The account created by the customer is stored in the database

Operation	Pick Slot
Preconditions	Customer has not picked a slot
Postconditions	The slot picked by customer is remembered by the system

Section 6: Project Size Estimation

https://www.tutorialspoint.com/estimation_techniques/estimation_techniques_use_case_points.htm

Use-Case Complexity	Number of Transactions	Use-Case Weight
Simple	≤3	5
Average	4 to 7	10
Complex	>7	15

Number of Simple use cases: 6
Number of average use cases: 5
Number of complex use cases: 2
Unadjusted uses case weight: 110

Actor Complexity	Example	Actor Weight
Simple	A System with defined API	1
Average	A System interacting through a Protocol	2
Complex	A User interacting through GUI	3

Number of Simple actors: 1
Number of Average actors: 2
Number of complex actors: 4
Unadjusted Actor weight: 17

Unadjusted Use case points = UUCW+UAW = 110+17 = 127

Factor	Description	Weight
T1	Distributed System	2.0
T2	Response time or throughput performance objectives	1.0
T3	End user efficiency	1.0
T4	Complex internal processing	1.0
T5	Code must be reusable	1.0
T6	Easy to install	.5
T7	Easy to use	.5
T8	Portable	2.0
T9	Easy to change	1.0
T10	Concurrent	1.0
T11	Includes special security objectives	1.0
T12	Provides direct access for third parties	1.0
T13	Special user training facilities are required	1.0

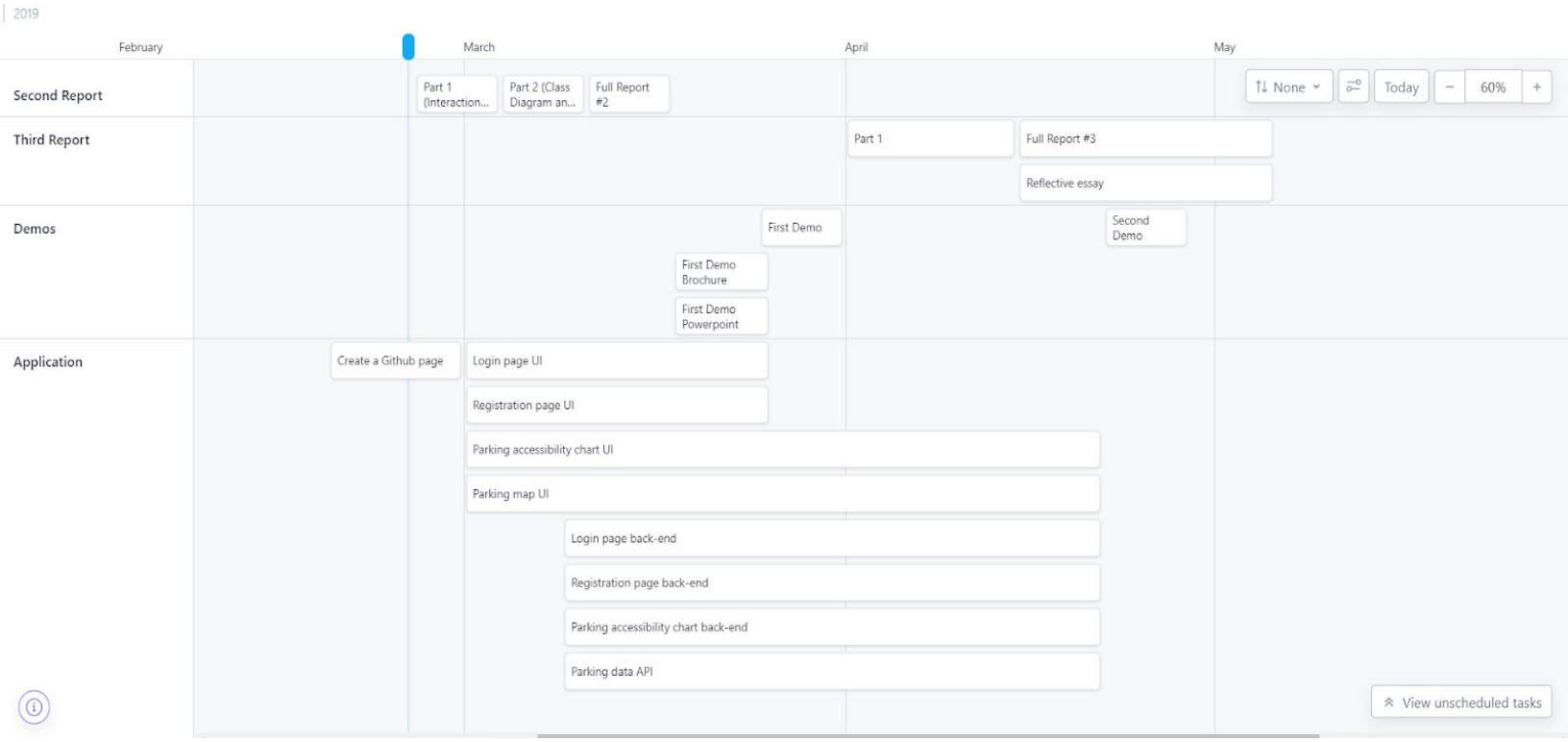
T1 Value: 2.5 T1 Impact: 5.0
T2 Value: 3.0 T2 Impact: 3.0
T3 Value: 3.0 T3 Impact: 3.0
T4 Value: 2.5 T4 Impact: 2.5
T5 Value: 4.0 T5 Impact: 4.0
T6 Value: 4.0 T6 Impact: 2.0
T7 Value: 4.0 T7 Impact: 2.0
T8 Value: 4.0 T8 Impact: 8.0
T9 Value: 2.0 T9 Impact: 2.0
T10 Value: 1.0 T10 Impact: 1.0
T11 Value: 0.5 T11 Impact: 0.5
T12 Value: 5.0 T12 Impact 5.0
T13 Value: 0 T13 Impact: 0
TFactor : 38
TCF = 0.6 x (.01 x TFactor) = 0.98

EF = 1

UCP = TCF x UUCP x EF = 130

Section 7: Plan of Work

After submitting report #1, we plan on working on working on report #2. We also plan to begin working on the programming aspect of the project. To start off, we would work on the UI aspect of code so that we could present how the application would look by the first demo. However, if time permits, we would also begin working on the back-end code for the application earlier on. This includes retrieving parking data from lots and data from students parking permit. We would also need to develop a database for the parking lots and the respective parking spaces. Later on in the future, we would also need to begin working on report #3 and the reflective essay. In addition, we would prepare for the second demo and finish up the application code.



<u>Product Ownership:</u>

- Registration page UI: Anthony Lau
- Login page UI: Josh LoGiudice
- Parking options page UI: Gabriel Shen
- Parking accessibility chart UI: Max Davatelis

- Parking map UI: Andrew Schneeloch
- Login page data (back-end): Jahidul Islam
- Registration page data (back-end): Krithika Uthaman
- Parking accessibility chart (back-end): Suva Shahria
- Parking data API (back-end): Yu Liu

Section 8: References

Banerjee, and Associates. “An Overview of Common Parking Issues, Parking Management Options,and Creative Solutions.” *Pipta*, pipta.org/wp-content/uploads/2014/04/Parking-Problems-and-Creative-Solutions.pdf

https://pipta.org/wp-content/uploads/2014/04/Parking-Problems-and-Creative-Solutions.pdf

Tutorialspoint.com. “Estimation Techniques Use-Case Points.” *Www.tutorialspoint.com*, Tutorials Point, www.tutorialspoint.com/estimation_techniques/estimation_techniques_use_case_points.htm.