Fitness and Nutrition Buddy Project

Report

Prepared by

Andy Hansana, Kevin Elliott, Vincent Weaver, Ryan Crowley
***For use in* CS 440**
at the
University of Illinois Chicago

September 2021

Table of Contents

Page left blank

List of Figures

List of Tables

I Project Description

1 Project Overview

Our application, Fitness and Nutrition Buddy, will be a great application utilized by those interested in fitness, as well as keeping track of their daily nutrition. This application will use the users current location and will be able to locate all of the nearest restaurants around them as well as using a filter for certain restrictions/cravings that they desire (for example low carbs, low fat, etc). This will benefit the user by allowing them to intuitively live a healthy lifestyle, while not having the hassle of searching for nearby restaurants. They also will not have to search through all of the results/each individual menu. It will also be able to keep track of the user's steps, and calories burned (similar to the Fitness app on the Apple Watch).

2 The Purpose of the Project

This project is being done for convenience for those who aspire or currently have a healthy fit life. With this application being developed, it will be much easier for someone to figure out what they would want to eat on a diet, or to accommodate for any dietary restrictions. This would target an audience that would be interested in having a healthier lifestyle, or even if any users are trying to lose weight or have restrictions on their diet.

2a The User Business or Background of the Project Effort

The business that would benefit from this application would be certain fitness companies such as Gymshark or Alphalete, however it could branch out to many other companies to benefit them such as Nike or any specific company interested in branching out and working with a fitness application. The business being doing would be users needing an application for their needs and conveniences and then them applying it to their life.

2b Goals of the Project

We want to provide a convenient accessible application to customers so that they can access foods around them that accommodate their diet and tracking their fitness all in the palm of their hands.

2c Measurement

We would like to provide a premium subscription of our application if the user refers 5 people that download and use our application, as well as providing those with a free trial of how it is to upgrade to premium.

3 The Scope of the Work

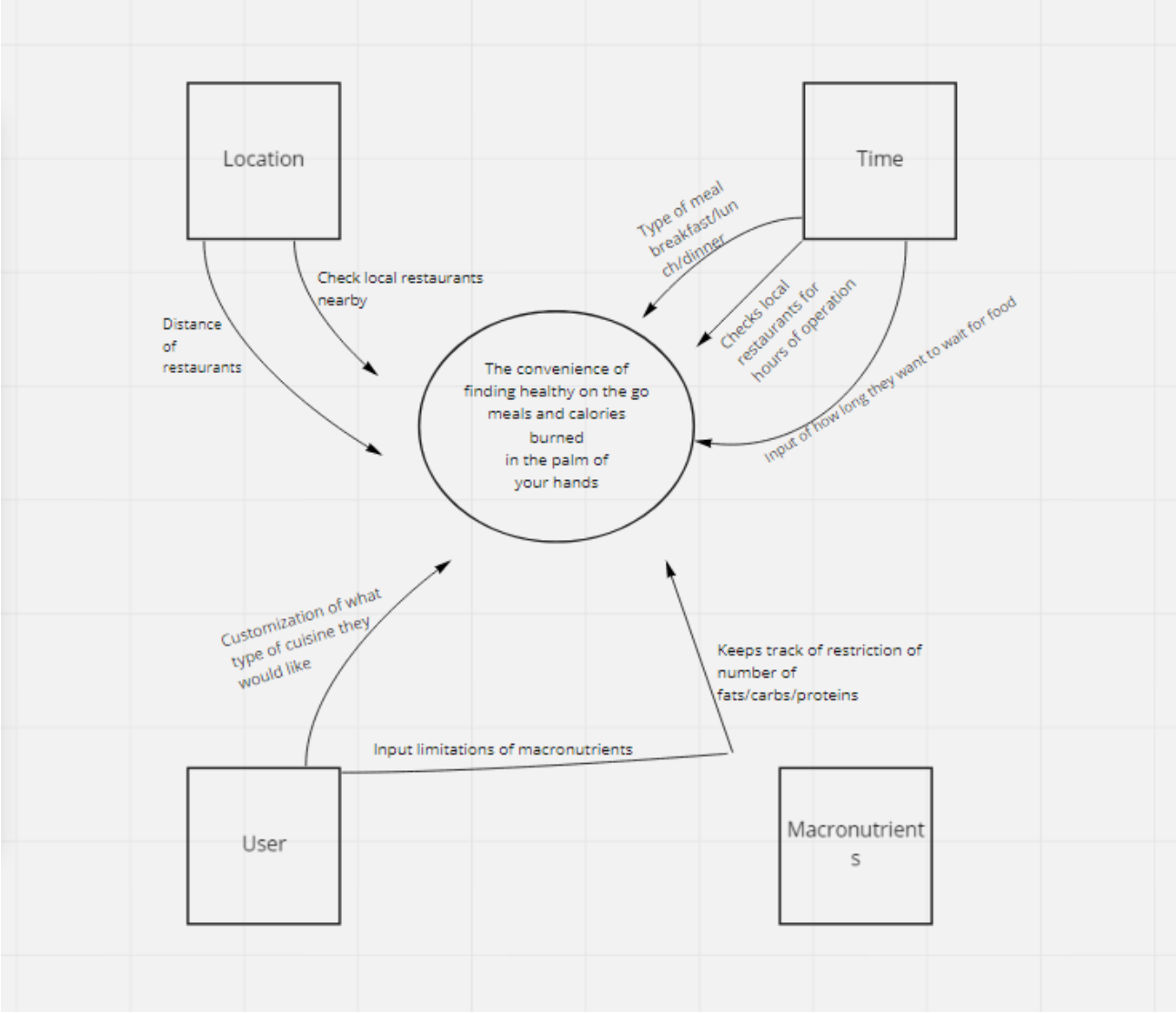
The work would be described as dietary nutrition needed for the users and the work would be providing the different dining options for the user available around them.

3a The Current Situation

The client conducting the work is the application being built in order for the users to be able to access it once it launches on the app store. Launching this business will incorporate updating the application daily and incorporating new restaurants in the area.

3b The Context of the Work

Page left blank



What is included within this application is the location, macronutrients, user inputs and location for the app to pull data and display it onto the interface which will consist of meals from different restaurants. The application has the capabilities of getting user customizations of what meals are liked and what aren't. What wouldn't be provided for this application would be certain restaurants that aren't specific on their macronutrient information for customers online.

3c Work Partitioning

Page left blank

Event Name	**Input Summary**	**Output Summary**
User burns calories	Application tracks number of calories burned from heartrate and movement	Calories burned output
User searches for food near them	Location and restaurants near them	Meals that they are able to buy on the go near them
Ketogenic foods near UIC	Location of UIC with	Ketogenic carbless foods

	restrictions of carbohydrates	such as (grilled chicken and vegetables) at Panda Express
Low Fat Snacks around the area	Takes the low fat restriction into consideration with x amount of fat and under	Snacks that are the same or less fat than requested fat such as greek yogurt
Input of a certain location where a restaurant is closed	Hours of operation for that certain restaurant	Gives message saying restaurant is closed and recommends somewhere else
User locates food near them at 5 AM with low calories	Location as well as late time	Only would output the lowest calorie meals at a late night restaurant/food area.
User updates their macros on their profile	User changes their macro goals or calorie goals for their daily counter	Notification is sent that the update was successful and the values are properly updated in the goals
User goes over calorie limit	User inputs a food that puts them over one of their goals for the day	Notification that gives tips on how to stay under and target the foods that contributed the most
User logs in for a streak	User logs in for the first time of the day	Every day the user logs in it is recorded on a calendar or counter
User breaks streak	User fails to login at any point in the day	Notification informing then on their streak and if they want to input what they ate yesterday

3d Competing Products

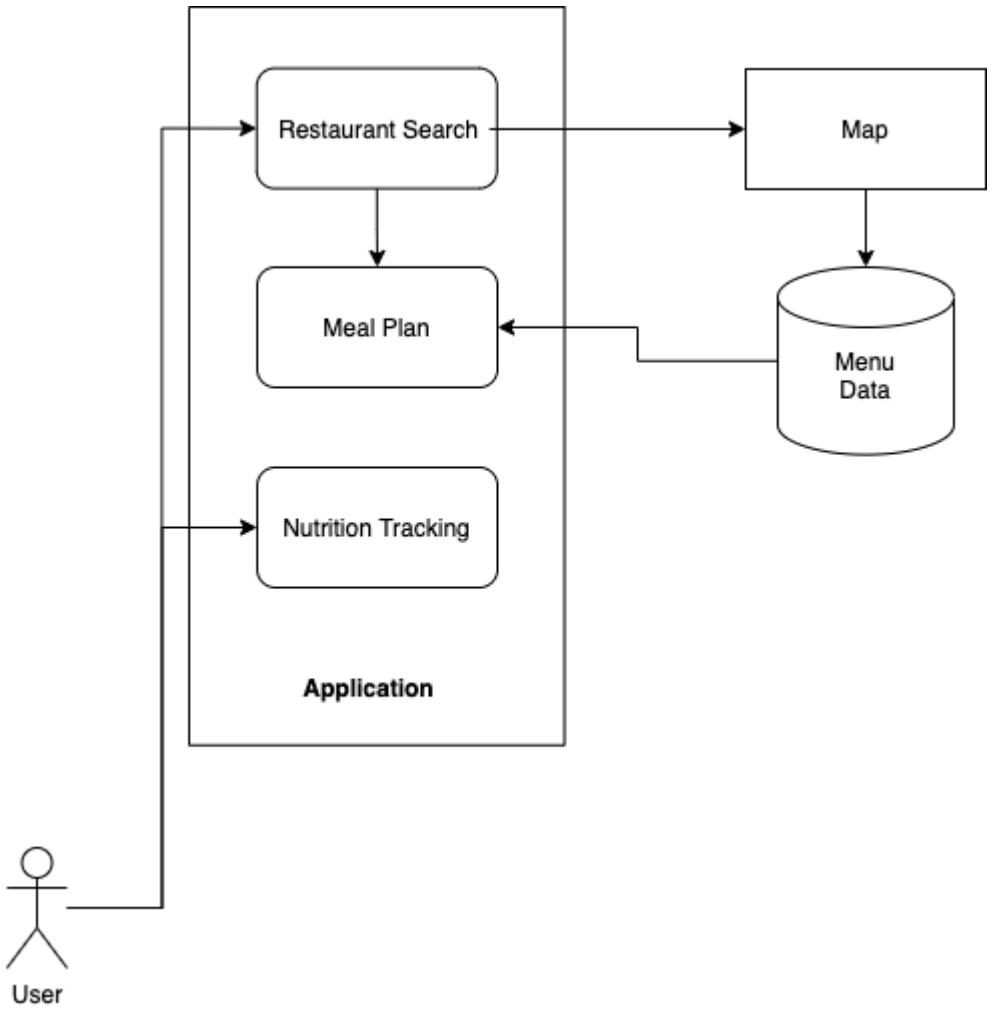
Other competitors that would compete in this market for this application would be MyFitnessPal, which is a fitness/nutrition mobile app that is capable of tracking your calories and fitness. One benefit that we would implement that our competitor doesn't provide is the ability to output what meals you would exactly be able to buy near you that fit your health goals. On top of that MyFitnessPal isn't always accurate as users are able to input calories of a certain food on the database that other users are able to access and read

which leads to data inaccuracy.

4 The Scope of the Product

The scope of our product would be the consistency and convenience of being able to provide healthy/convenient on the go meals that fit in our users dietary restrictions with them not having to stress over researching restaurants in their area and calculating all of the macronutrient information required for their diets. What is included in this application is tracking meals from restaurants around you that you're able to pick up and grab on the go and a tracker to see the amount of calories you burned. What isn't included on this application is tracking calories of meals that are from restaurants that do not provide nutritional information of their menus online.

4a Scenario Diagram(s)



4b Product Scenario List

Scenario Name	*Participating Actors*
1. *Nutrition Tracking*	*Fitness and Nutrition Buddy users*
2. *Meal Plan*	*Fitness and Nutrition Buddy users, Developers*
3. *Restaurant Search*	*Fitness and Nutrition Buddy users*
4. *Map*	*Fitness and Nutrition Buddy users, Google*
5. *Menu Data*	*Developers, Restaurants*

4c Individual Product Scenarios

- Nutrition Tracking: The user is able to track their calories and nutrition data on a day-to-day basis using the app's interface. This will be one of the main screens that the user interacts with when they load into the Fitness and Nutrition Buddy.
- Meal Plan: The user can create meal plans based on the restaurants they search for. This

will consist of pulling menu data from different restaurants and calculating calories, as well as sorting foods by their nutrition groups. This portion of the app will also filter any specifications the user may input based off of their current diet.

- Restaurant Search: A search feature is available for the user to find specific restaurants in their general vicinity, which will be connected to Google Maps. The user is able to input preferences like the type of food or the general location.
- Map: The app utilizes Google Maps to display a map to the user, which will show various restaurants around their location. The app will display the user's search results with this map by indicating the restaurants' locations, as well as a summary of some relevant data based on their search specifications.
- Menu Data: Data is pulled from Google pertaining to menu information, which is sent to the meal plan portion of the app in order to do the calculations for optimizing a meal for the user.

5 Stakeholders

5a The Client

The development organization would be the initial clients, working towards the goal of eventually selling the product to a large company in the fitness industry, similar to under armor owning myfitnesspal.

5b The Customer

The customer would be a big fitness brand that is looking to expand into a mobile market, some brands that could possibly be interested would be gym shark, nike, weight watchers, gnc. A brand that has an established presence in the fitness industry with or without a mobile app presence.

5c Hands-On Users of the Product

The user could be anyone on the app stores of an iphone or android, with the widespread availability of the apps in the app store the potential customers are very large. It would be up to the client if they want to charge the customers up front or via premium services for a subscription. The target demographic of the customers is those who workout and incorporate meal plans, people who want to lose weight, or even anyone that just wants to find new meals from restaurants in the area. The users are responsible for inputting their goals and calories eaten, and requesting meals from the app. The user's knowledge can be novice since the app is designed to do the harder parts of macro management for them, but being an expert does not harm the user's experience. The app will not have any particularly hard to navigate portions so anyone who can competently use their smartphone will be able to use

the app. The users can be fit or unfit, and anywhere in between. Disabilities are not relevant as long as the user is able to go out to eat. Users would typically be anyone that works out or cuts calories so teenagers and up. Education and linguistic skills are not needed besides basic english. Users' attitude towards technology helps with the app as being more connected to your phone means more opportunities to interact with the app based on the food consumed.

5d Maintenance Users and Service Technicians

SV: Describe users that will install, maintain, update, and otherwise service the product as needed. May not apply to all projects.

Content

Maintenance users are a special type of hands-on users who have requirements that are specific to maintaining and changing the product.

Motivation

Many of these requirements will be discovered by considering the various types of maintenance requirements detailed in section 14. However, if we define the characteristics of the people who maintain the product, it will help to trigger requirements that might otherwise be missed.

No physical installation or maintenance is required besides server upkeep which would be handled by the updates team. The team that maintains the product would handle any of the servers needs along with and addition features the client wants.

5e Other Stakeholders

Other stakeholders that this application would need to include would be:

- Testers (for debugging purposes)
    - Knowledge needed for test cases and debugging code
    - Testers would be semi-involved as a bug could potentially prohibit users from using the application until fixed
    - Working together with Original developer team for combining knowledge for promotion of growth for the app.
- Marketing
    - Sponsorships towards influencers (better advertising for the product)
    - Important involvement as marketing is a huge factor for sales and promotion
    - Huge influence on success rates if advertised well
- Legal Team
    - Knowledgeable of lawsuits as well as dealing with any terms of the law if needed
    - Will be important as if any other companies try to copy/rebrand our own

unique product
- Significant impact and reliability on the company as if there is no legal team, company can get sued and shut down.

5f User Participation

23

Users can expect to participate in the development of the app by using a beta version of it enough features have been added. Whether the beta will be free or paid for is up to the client but there would be temporary features involved to poll users experience in the app so far to provide feedback.

5g Priorities Assigned to Users

24

The key users are those that are fit and looking to maintain a diet with exact macros and also those who are trying to lose weight through a diet. These are the two groups that have the most to gain by consistent use and the features should be prioritizing them. Secondary users would be those who use the app on a semi consistent basis and are generally into fitness but not fully committed. These users are the most likely to grow into the key user group as time goes on so their needs should also be considered so the biggest part of the userbase is cared for. The unimportant users are the ones that use the app for new food recommendations, the app could certainly be used in this way but it is not the intended use, therefore if users are unhappy with the focus on fitness and not just focusing on new meals from restaurants their concerns don't have much weight to the team.

6 Mandated Constraints

6a Solution Constraints

25

Description: The final product should be a mobile application, or at least have its main platform be a mobile application.

Rationale: This must be done to provide the user with on-the-go capability. The user should be able to obtain healthy meal options wherever they are.

Fit criterion: The app must be available on the Apple App Store and Google Play for users to download, and should provide the user with a view of their current location on startup.

Description: The final product should provide accurate nutritional information and

calculations for the user's personal dietary specifications.

Rationale: The app needs to provide an effective means of improving the fitness and nutrition of the user.

Fit criterion: The app must utilize Google Maps data, or menu data directly from the restaurant only in order to form accurate nutritional calculations.

Description: The app must provide a quick way to find meals for the user.

Rationale: The user may be in certain social situations where decisions need to be made quickly (i.e. at a sit-down restaurant). Also, this is for the user's convenience when trying to find healthy options.

Fit criterion: The time frame for searching for restaurants and meals must be under 30 seconds.

6b Implementation Environment of the Current System

The product will be designed as a mobile application that will be catered towards the user installing on their personal device. The application will run on all major mobile platforms (i.e. IOS, Android, etc.).

6c Partner or Collaborative Applications

There are an extreme number of restaurants around the world with different data and information regarding their menu items. Collaborating with applications such as Google Maps, Apple Maps, and Waze, as well as restaurant databases and APIs such as Nutritionix and Chomp would be useful. Their location services would increase the speed and efficiency of the app, and the API data would allow for accurate and informative data to provide to the user.

6d Off-the-Shelf Software

Content

Some software that must be included in the final product includes a 24 hour server that updates real time restaurant data regularly. The server software must also store data from the users and build specific diets based on prior food results. Having this 24 hour server would also include physical hardware that continuously runs the server.

Motivation

Ideally, other products that are being looked at to be incorporated into the final product include some potential calorie tracking software in other devices.

Examples

Utilizing a smart watch from companies like Fitbit or Apple's watch product is in consideration when building this product. The application would really benefit from adapting to a separate mobile device such as smart watches since it can be more accessible and convenient for users than a phone.

Considerations

The immediate conflict with attempting to get the OTS software working with the Fitness and Nutrition Buddy is compatibility with the smart watch device and communicating with the companies who own these devices.

6e Anticipated Workplace Environment

Content

The product is designed to be used while the user is both stationary or on the move. Ideally, users have the flexibility to be either at home or away while using the application. Especially when users are out shopping or getting out of the gym, being able to use the app anywhere is core to its functionality.

Motivation

The application uses data from satellite maps and restaurants and combines both data sets to feed the user relevant information about where and what to eat based on the users choice of food. Allowing the application to be compatible with smart watches as well as mobile devices allows for flexibility in user workspace.

Examples

Some examples of applications for satellite map data/APIs are from Google Maps, Waze, or Apple's map data. Some examples of programs that utilize food chains are Uber Eats, GrubHub, etc. These applications are examples that allow for both mobile use or stationary use.

Considerations

The developers would be expected to work in person for the majority of the week. Whether or not a hybrid system like many other companies would be up to the team based on how they have been working post start of development. If they choose to go hybrid, employees are expected to supply their own devices.

6f Schedule Constraints

There are not any hard deadlines for the app in the development phase, and the best timing for it is a mix of waiting to release as the budget allows it. Given the pandemic is not gone away yet some fast food locations are operating differently and the longer the app takes to release the more normal things will become. Obviously the team doesn't have the budget to wait forever but if there were to be a large set back the app does not fail. Having to miss the release for this calendar year and take Q1 of the next is not ideal but still doable.

6g Budget Constraints

Since the project aims to get bought out by a large company eventually, it is in the team's best interest to create a quality product as fast as possible to maximize the use of the budget. There wouldn't be any budget cuts exactly since no company will buy a cheaply made app, but the emphasis for the budget would instead be on avoiding setbacks if possible to make the time of development smaller. The resources for the product would be the developers and the team.

7 Naming Conventions and Definitions

7a Definitions of Key Terms

Meal: a combination of various menu items provided to the user after requesting a food recommendation from the app.

Calories: how many calories a food item has

MealCalories: how many calories an entire meal has

Carbs: one of the 3 macros the app will track, carbohydrates, refers to anything that falls under the carbohydrates section of a nutrition label

Fat: one of the 3 macros the app will track, fat, refers to anything that falls under the fat section of a nutrition label

Protein: one of the 3 macros the app will track, protein, refers to anything that falls under the protein section of a nutrition label

Macros: all three of carbs fat and protein together make up a users macros that they set

Restaurant: a place that sells fast food, dine in, take out etc. does not refer to grocery stores

Streak: user logging in for a x amount of days

Location: Where the user currently is, particularly when they request for a meal

Profile: where the users attributes are stored such as macros, streak, etc

User: current instance of the app

7b UML and Other Notation Used in This Document

This document generally follows the Version 2.0 OMG UML standard, as described in, M. Fowler, UML Distilled, Third Edition, Boston: Pearson Education, 2004. Exceptions are noted in their specific cases.

7c Data Dictionary for Any Included Models

33

DailyTotal = totalCalories and totalCarbs and totalFat and totalProtein

Meal = 1 or more menu items

MealCalories = sum of calories from every item in a meal

8 Relevant Facts and Assumptions

8a Facts

67% of americans are overweight or obese
84.8 million people eat fast food a day in the US
Application will be made in the appropriate language for the platform(swift for iOS)
19.3% of americans are participating in some form of exercise each day
The average american consumes 3600 calories a day
The recommended daily caloric intake is 2000 for women and 2500 for men

8b Assumptions

34

Page left blank

35

Assuming fast food chains will either continue to operate at current capacity or improve

Assuming all developers will be able to code in all mobile environments

Assuming servers will either be hosted via a reliable database or in house

Assuming google maps api access is maintained

Assuming once a client purchases the product the team is staying with the product

Assuming covid does not worsen and create another lockdown

II Requirements

SV: Sections 9 and 10 deal with functional requirements. Sections 11 to 20 are a very thorough list of possible non-functional requirements, not all of which apply to every project. You should think carefully about each of these, form requirements if applicable, or write "Not Applicable" otherwise. See section 10 for the format of individual requirements. Section 21 documents the acceptance tests planned to verify the requirements – See that section for further details, and be aware that every requirement needs at least one verifying acceptance test (though some tests may verify more than one requirement.)

9 Product Use Cases

SV: Product Use Cases are very similar to Product Scenarios, but in more formal detail. They serve as a first step towards developing functional requirements, and can aid in organizing requirements according to the use case(s) from which they were developed. See the CS 440 web site for a sample use-case form, with instructions.

This section begins to describe in more specific and precise detail exactly what steps the system takes in the course of its performance. Use cases serve not only to more specifically define the system (and its boundaries), but also to identify functional requirements, to identify initial objects / classes, and to organize the work.

9a Use Case Diagrams

SV: Use case diagrams list the use cases developed for a system, mark the boundary of what is internal or external to the system to be developed, and indicate which external entities (actors) are associated with each use case.

Use Case diagrams serve two purposes: As a form of graphical table of contents listing the individual use-cases, and also to define the boundary of what is included as part of the proposed system and what is not included.

A use case diagram identifies the boundaries between the users (actors) and the product. You arrive at the product boundary by inspecting each business use case and determining, in conjunction with the appropriate stakeholders, which part of the business use case should be automated (or satisfied by some sort of product) and what part should be done by the user. This task must take into account the abilities of the actors (section 3), the constraints (section 4), the goals of the project (section 1), and your knowledge of both the work and the technology that can make the best contribution to the work.

The use case diagram shows the actors outside the product boundary (the rectangle). The product use cases are the ellipses inside the boundary. The lines denote usage.

Note that actors can be either automated or human.

Depending on the complexity of the product it may be necessary to use more than one diagram to list all of the use cases. When more than one diagram is required the use cases can be divided up several ways: Normal operations versus exceptional cases, or daily tasks versus monthly tasks, or user tasks versus administration tasks, etc.

Examples

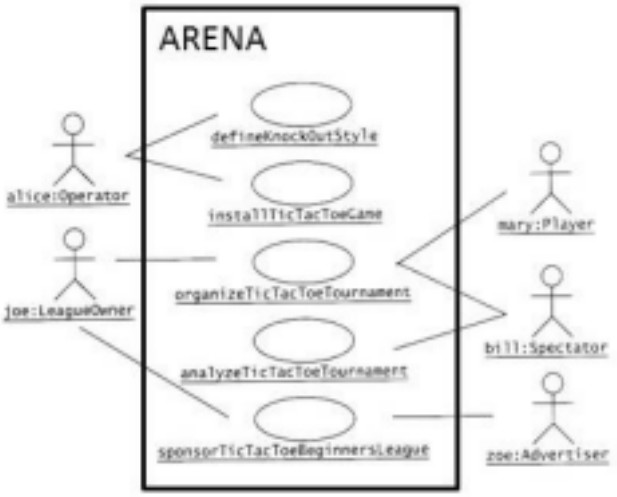


Figure 1 - Sample Use Case Diagram from Bruegge & DuToit (modified)

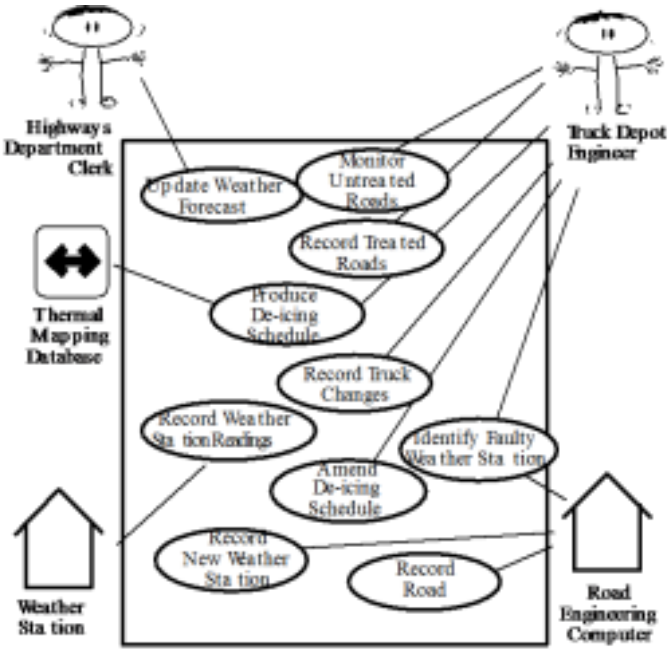


Figure 2 - Sample Use Case Diagram from Robertson and Robertson

Derive the product use cases by deciding where the product boundary should be for each business use case. These decisions are based on your knowledge of the work and the requirements constraints.

9b Product Use Case List

SV: A list (table) of use cases is an alternative to the use case diagram, particularly when there are many use cases. There may be additional information in the table not found in the diagram, such as cross referencing to other sections or materials.

The use case diagram is a graphical way of summarizing the product use cases relevant to the product. If you have a large number of product use cases (we find 15–20 is a good limit), then it is better to make a list of the product use cases and model or describe each one individually.

9c Individual Product Use Cases

Use cases are similar to scenarios, in that both tell the story of how the system interacts with the user(s) in response to some business event or while conducting some business task. The difference is that use-cases are much more formal, with certain pre determined sections for each use-case, and that use-cases indicate clearly what action the system takes in response to what actions taken by the user.

SV: The following example was copied from "useCaseFormWithInstructions.docx", available on the CS 440 web site. (There is also a blank version available.)

Use case ID: Name: pre-conditions: post-conditions: Initiated by: Triggering Event: Additional Actors:
Sequence of Events: 1. Initiating event or action should be step 1, taken by initiating actor. 2. System response follows, indented right. 3. All external action steps are aligned with step 1. ("stimulus" style) 4. All system responses are indented right, aligned with step 2. ("response" style) 5. All steps should be expressed in the active voice, clearly indicating **who** performs each action 6. The sequence of events should show a back-and-forth stimulus response relationship.
Alternatives: These would be normal and expected variations from the base case. Exceptions: These would be unusual variations from the base case, often caused by problems.

- *For all of the above, list as NA if not applicable.*
- *The following may be added if relevant, or omitted otherwise:*
  - *related use cases or scenarios*
  - *associated tests, systems, classes, etc.*
  - *revision history*
  - *references to other documents*
  - *author(s) / originator(s)*

- *notes*

• Alternatives and Exceptions may be listed either as separate use cases or as notes to a base case, depending on their significance and similarity.

• For regularly occurring periodic events, "time" can be listed as the initiating actor.

You may also want to view Figure 4.7 from "Object Oriented Software Engineering" by Bruegge and DuToit

10 Functional Requirements

SV: Each requirement listed needs to have a unique identifier, a short name, a one- or two-sentence description, a rationale, a fit criteria, and reference to one or more acceptance tests to be used to confirm the completion of this particular requirement. The acceptance tests themselves are documented in section 0- See that section for further details. It is recommended to number the requirements according to their type, such as F-4 for the fourth functional requirement or U-2 for the second usability requirement. Functional requirements specifically deal with the functionality the system must have, and are generally derived directly from the steps the system takes during use cases.

SV:

001 - Diet Style

Description: This is where the user will input their style of calorie and macro restrictions so that the app can proceed to make their customized menus
Rationale: The app cannot make any custom meals without the restraints on it

Fit Criterion: Does the user's profile have its custom settings prompted on start up for the first time and is it saved upon relaunch, along with being able to change at any time.

Acceptance Tests: Save user data, input diet data

002 - Location Services

Description: When the user requests a meal from the app, it needs to know where it is and what's around, pulling in data from google maps will solve both
Rationale: The apps primary function is based on location

Fit Criterion: When the user requests food does location services draw where they are and also the restaurants around them

Acceptance Tests: Location Loads, restaurants load

003 - Meal Creation

Description: Drawing from the menus of restaurants around user, the app will create a meal for them that is restrained by the users calorie limit along with macro limits. After the meals are created they are added to a table along with every other meal combo from the other restaurants
Rationale: The app does not need any advanced filtering for these meals since its purpose is to show the users all of their options even if one is not nearly as satisfying as the others

Fit Criterion: When the user requests meals, a table populates with meal combos based of user limits from the restaurants around them

Acceptance Tests: Meal Creation, Meals under set limits, Meals from all restaurants

Content

A specification for each functional requirement. A full explanation is included in this template's introductory material.

Motivation

To specify the detailed functional requirements for the activity of the product.

Fit Criterion

Each functional requirement should have a fit criterion or a test case. In any event, the fit criterion is the benchmark to allow the tester to determine whether the implemented product has met the requirement.

Considerations

If you have produced an event/use case list (see sections 7b and 8a), then you can use it to help you trigger the functional requirements for each event/use case. If you have not produced an event/use case list, give each functional requirement a unique number and, to help with traceability, partition these requirements into event/use case–related groups later in the development process.

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

Acceptance Tests: List ID# and/or names here . . .

11 Data Requirements

SV: Data requirements deal with requirements that are somehow related to data, such as the definition of what is included in a "student record" or the acceptable form of an e-mail address or allowable range of certain data items.

SV: Data requirements: User profile data containing nutritional information, location services, restaurant menu information

Content

A specification of the essential subject matter, business objects, entities, and classes that are germane to the product. It might take the form of a first-cut class model, an object model, or a domain model. Alternatively, these requirements might be described by defining the terms in the dictionary described in section 5.

Content: User class, Restaurant class, Map object, Meal object

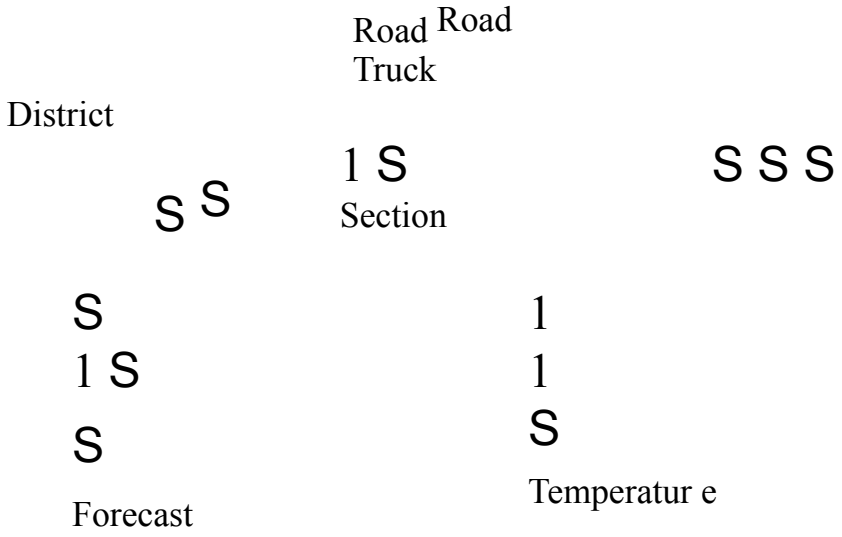
Motivation

To clarify the system's subject matter, thereby triggering recognition of requirements not yet considered.

Motivation: User should be able to login and based on their macros get a meal from surrounding restaurants

Example

This is a model of the system's business subject matter using the Unified Modeling Language (UML) class model notation.



You can use any type of data or object model to capture this knowledge. The issue is to capture the meaning of the business subject matter and the connections between the individual parts, and to show that you are consistent within your project. If you have an established company standard notation, use that, as it will help you to reuse knowledge between projects.

Considerations

Are there any data or object models for similar or overlapping systems that might be a useful starting point? Is there a domain model for the subject matter dealt with by this system?

Considerations: Google maps is the most likely way to get location services, and many times they use Yelp for menus since restaurant owners put their own on there very commonly.

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

Acceptance Tests: List ID# and/or names here . . .

12 Performance Requirements

12a Speed and Latency Requirements

SV: Requirements specifying how fast (or slow) the product must operate or how much lag is allowable between stimulus and either initial response or task completion. Other timing-related requirements could go in this section.

SV: Any time a user presses a button there needs to be some kind of feedback that input was received. If it is a process that takes longer than a couple seconds then a loading indicator of some kind should be in place. On long tasks that take more than

10 seconds, a loading bar with updates should be in place.

Content

Specifies the amount of time available to complete specified tasks. These requirements often refer to response times. They can also refer to the product's ability to operate at a speed suitable for the intended environment.

SV: accepting user input should be instant. Getting users location should be under 5 seconds from when they allow it. Loading restaurants menus should be no longer than 1 minute and creation and loading of meals should be no longer than 2 minutes

Motivation

Some products—usually real-time products—must be able to perform some of their functionality within a given time slot. Failure to do so may mean catastrophic failure (e.g., a ground-sensing radar in an airplane fails to detect an upcoming mountain) or the product will not cope with the required volume of use (e.g., an automated ticket selling machine).

Motivation: The app has an emphasis on speed but ultimately is not the highest priority, when someone is out and looking for something to eat they do not want to wait more than a few minutes to figure out what they want. Therefore the app should be able to complete its function from requesting a meal to displaying it in around 2-3 minutes at most, the faster the better. Users taking time to decide what they want is not factored into the time.

Examples: User pressing a button means the button moves or highlights on press and the function is carried out within 2 seconds. Response needs to be fast enough to not be distraction or interrupting. The product should update location services every 10 seconds so that if they user in a vehicle it stays accurate.

Examples

Any interface between a user and the automated system shall have a maximum response time of 2 seconds.

The response shall be fast enough to avoid interrupting the user's flow of thought.

The product shall poll the sensor every 10 seconds.

The product shall download the new status parameters within 5 minutes of a

change. Fit Criterion

Fit criteria are needed when the description of the requirement is not quantified. However, we find that most performance requirements are stated in quantified terms. The exception is the second requirement shown above, for which the suggested fit

criterion is

The product shall respond in less than 1 second for 90 percent of the interrogations. No response shall take longer than 2.5 seconds.

Fit criterion: the creation of meals should take no longer than 3 minutes

Considerations

There is a wide variation in the importance of different types of speed requirements. If you are working on a missile guidance system, then speed is extremely important. By contrast, an inventory control report that is run once every six months has very little need for a lightning-fast response time.

Customize this section of the template to give examples of the speed requirements that are important within your environment.

Considerations: The users will be hungry so making them wait is not ideal, but not the end of the world, the process shouldnt make users feel like they are sitting there waiting around when they could just go get a meal before the app even shows anything.

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

Acceptance Tests: List ID# and/or names here . . .

12b Precision or Accuracy Requirements

SV: Self-explanatory. How accurate or precise must the system be.

SV: The location services does not have to be very accurate, so anywhere within 100 feet of the user is fine as an achor point. The meal creation needs to be hard limited at the users set restrictions.

Content: People are counting ever calorie and would not have any use for a meal thats “only” 50 cals over their limit.

Motivation: People are using the app to lose weight or gain muscle

and both need strict rules and attention to detail to get right

Content

Quantification of the desired accuracy of the results produced by the product.

Motivation

To set the client's and users' expectations for the precision of the product.

Examples

All monetary amounts shall be accurate to two decimal places.

Accuracy of road temperature readings shall be within ±2°C.

Considerations

If you have done any detailed work on definitions, then some precision requirements might be adequately defined by definitions in section 5.

You might consider which units the product is intended to use. Readers will recall the spacecraft that crashed on Mars when coordinates were sent as metric data rather than imperial data.

The product might also need to keep accurate time, be synchronized with a time server, or work in UTC.

Also, be aware that some currencies have no decimal places, such as the Japanese yen. **ID# - Name**

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

Acceptance Tests: List ID# and/or names here . . .

12c Capacity Requirements

SV: Requirements regarding the largest "thing" the system must be able to handle, or perhaps how many things it can handle (at once.) Note: Requirements regarding how many things it can handle in a given time period would be a speed requirement, covered in section 12a above.

SV: Largest thing the app will handle is location and restaurant menus, with the smaller things being user profile and meal table

Content: App will have to download map data and restaurants around it which is an estimated 50MB every other part of the app is estimated to be 100MB

Motivation: This is a mobile app so every MB counts towards more users being able to download the app

Content

This section specifies the volumes that the product must be able to deal with and the amount of data stored by the product.

Motivation

To ensure that the product is capable of processing the expected volumes.

Examples

The product shall cater for 300 simultaneous users within the period from 9:00 A.M. to 11:00 A.M. Maximum loading at other periods will be 150 simultaneous users.

During a launch period, the product shall cater for a maximum of 20 people to be in the inner chamber.

Fit Criterion

In this case, the requirement description is quantified, and thus can be tested.

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

Acceptance Tests: List ID# and/or names here . . .

13 Dependability Requirements

13a Reliability Requirements

SV: Reliability relates to how frequently the system fails, (either by shutting down or by delivering erroneous results), and the consequences of those failures. These requirements may also address the conditions under which it is allowed to fail (or not.), See also availability and robustness in the following sections.

SV: Under no circumstances should the app fail to load user data, location services can fail if the users connection isn't strong enough or disconnected, same with restaurants. Given the past two load, the creation and delivering of meals should not ever fail to get to the user

Content: if the users connection is fine then there should never be a failure to laid, the only reason it should happen is if the connection is poor or offline

Motivation: If the app fails too often then users will leave, but its not a major emergency if it does fail so occasionally if it does it is alright if the users connection is not consistent. The product should not fail more than once a day, in the event of a failure the users data needs to be preserved above all else.

Considerations: The product needs to be reliable so that at any given time if someone is hungry they can use it, but doesnt have many critical functions that cant fail no matter what, therefore the functionality should be what a user expects for any app.

Content

This section quantifies the necessary reliability of the product. The reliability is usually expressed as the allowable time between failures, or the total allowable failure rate.

Motivation

It is critical for some products not to fail too often. This section allows you to explore the possibility of failure and to specify realistic levels of service. It also gives you the opportunity to set the client's and users' expectations about the expected frequency and significance of potential failures.

Examples

The product shall not fail more than once per day.

No data shall be lost or damaged in the event of a failure. (This is an example of a ***fail-safe*** *requirement, which states that the product is allowed to fail, but it must do so safely.)*

Considerations

Consider carefully whether the real requirement for your product is that it is available for use or that it does not fail at any time.

Consider also the cost of reliability and availability, and whether it is justified for your product.

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

Acceptance Tests: List ID# and/or names here . . .

13b Availability Requirements

SV: Availability addresses the amount of time the system is running and available for use. It is affected by how often the system goes down (reliability), but also by the time required to bring the system back up again, the availability lost due to regularly scheduled maintenance down times, and the ability of the system to offer at least partial functionality in the face of failures or resource shortages. See also reliability and robustness.

Content

This section quantifies the necessary availability of the product. The availability is usually expressed as the fraction of total time that the system is up and available for use.

Availability is a function of the mean time between failures, the mean time required to bring the system back up after a failure, and the mean time the system is expected to be down for routine maintenance.

Motivation

There is a subtle distinction between how often a system goes down (reliability) and how much total time it spends being down (availability). This section allows you to specify realistic expectations about the amount of time that the product will be available for use.

Examples

The product shall be available for use 24 hours per day, 365 days per year. The

product shall be available for use between the hours of 8:00 A.M. and 5:30 P.M.

The escalator shall run from 6 A.M. until 10 P.M. or the last flight arrives. The product shall achieve 99 percent uptime.

Considerations

Consider carefully whether the real requirement for your product is that it is available for use or that it does not fail at any time.

Consider also the cost of reliability and availability, and whether it is justified for your product.

The sections on reliability and availability can sometimes be combined. **ID# - Name**

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

Acceptance Tests: List ID# and/or names here . . .

13c Robustness or Fault-Tolerance Requirements

SV: This section deals with the system's ability to provide at least partial functionality in the face of failures or resource shortages, such as operating in offline mode when network connectivity is unavailable. See also reliability and availability.

Content

Robustness specifies the ability of the product to continue to function under abnormal circumstances.

Motivation

To ensure that the product is able to provide some or all of its services after or during some abnormal happening in its environment.

Examples

The product shall continue to operate in local mode whenever it loses its link to the central server.

The product shall provide 10 minutes of emergency operation should it become disconnected from the electricity source.

Considerations

Abnormal happenings can almost be considered normal. Today's products are so large and complex that there is a good chance that at any given time, one component will not be functioning correctly. Robustness requirements are intended to prevent total failure of the product.

You could also consider disaster recovery in this section. This plan describes the ability of the product to reestablish acceptable performance after faults or abnormal happenings.

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

Acceptance Tests: List ID# and/or names here . . .

13d Safety-Critical Requirements

SV: These requirements address potential harm to health, safety, or property, and may refer to relevant standards such as OSHA compliance.

Content

Quantification of the perceived risk of damage to people, property, and environment. Different countries have different standards, so the fit criteria must specify precisely which standards the product must meet.

Motivation

To understand and highlight the damage that could potentially occur when using the product within the expected operational environment.

Examples

The product shall not emit noxious gases that damage people's health.

The heat exchanger shall be shielded from human contact.

Fit Criterion

The product shall be certified to comply with the Health Department's standard E110-98. It is to be certified by qualified testing engineers.

No member of a test panel of [specified size] shall be able to touch the heat exchanger. The heat exchanger must also comply with safety standard [specify which one].

Considerations

The example requirements given here apply to some, but not all, products. It is not possible to give examples of every variation of safety-critical requirement. To make the template work in your environment, you should customize it by adding examples that are specific to your products.

Also, be aware that different countries have different safety standards and laws relating to safety. If you plan to sell your product internationally, you must be aware of these laws. A colleague has suggested that for electrical products, if you follow the German standards, the largest number of countries will be supported.

If you are building safety-critical systems, then the relevant safety-critical standards are already well specified. You will likely have safety experts on your staff. These experts are the best source of the relevant safety-critical requirements for your type of product. They will almost certainly have copious information that you can use.

Consult your legal department. Members of this department will be aware of the kinds of lawsuits that have resulted from product safety failure. This is probably the best starting place for generating relevant safety requirements.

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

Acceptance Tests: List ID# and/or names here . . .

14 Maintainability and Supportability Requirements

14a Maintenance Requirements

SV: This section deals with the ease with which the system can be maintained, and possibly who will perform system maintenance and under what conditions. The ease of evolving the system into future versions may also be addressed here, or in a separate section (not included in this template) if that is a major concern.

Content

A quantification of the time necessary to make specified changes to the product.

Motivation

To make everyone aware of the maintenance needs of the product.

Examples

New MIS reports must be available within one working week of the date when the requirements are agreed upon.

A new weather station must be able to be added to the system overnight.

Considerations

There may be special requirements for maintainability, such as that the product must be able to be maintained by its end users or by developers who are not the original developers. These requirements have an effect on the way that the product is developed. In addition, there may be requirements for documentation or training.

You might also consider writing testability requirements in this section.

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

Acceptance Tests: List ID# and/or names here . . .

14b Supportability Requirements

SV: What ongoing support is to be provided, e.g. through a help desk. See also training requirements in section 16g below.

Content

This specifies the level of support that the product requires. Support is often provided via a help desk. If people will provide support for the product, that service is considered part of the product: Are there any requirements for that support? You might also build support into the product itself, in which case this section is the place to write those requirements.

Motivation

To ensure that the support aspect of the product is adequately specified.

Considerations

Consider the anticipated level of support, and what forms it might take. For example, a constraint might state that there is to be no printed manual. Alternatively, the product might need to be entirely self-supporting.

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

Acceptance Tests: List ID# and/or names here . . .

14c Adaptability Requirements

SV: Description of other platforms or environments to which the product must be ported.

Content

Description of other platforms or environments to which the product must be ported.

Motivation

To quantify the client's and users' expectations about the platforms on which the product will be able to run.

Examples

The product is expected to run under Windows XP and Linux.

The product might eventually be sold in the Japanese market.

The product is designed to run in offices, but we intend to have a version running in restaurant kitchens.

Fit Criterion

Specification of system software on which the product must operate.

Specification of future environments in which the product is expected to operate.

Time allowed to make the transition.

Considerations

Question your marketing department to discover unstated assumptions that have been made about the portability of the product.

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

Acceptance Tests: List ID# and/or names here . . .

14d Scalability or Extensibility Requirements

SV: The ease of expanding the system to a larger capacity as the business grows.

Content

This specifies the expected increases in size that the product must be able to handle. As a business grows (or is expected to grow), our software products must increase their capacities to cope with the new volumes.

Motivation

To ensure that the designers allow for future capacities.

Examples

The product shall be capable of processing the existing 100,000 customers. This number is expected to grow to 500,000 customers within three years.

The product shall be able to process 50,000 transactions per hour within two years of its launch.

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

Acceptance Tests: List ID# and/or names here . . .

14e Longevity Requirements

SV: This specifies the expected lifetime of the product.

Content

This specifies the expected lifetime of the product.

Motivation

To ensure that the product is built based on an understanding of expected return on investment.

Examples

The product shall be expected to operate within the maximum maintenance budget for a minimum of five years.

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

Acceptance Tests: List ID# and/or names here . . .

15 Security Requirements

SV: Security requirements address who is allowed what type of access to the system, and what areas require special protection or diligence. In practice security requirements must often be written by security experts, and may refer to standards.

15a Access Requirements

SV: These requirements address who has access to what (data or functionality) and under what conditions or restrictions.

Content

Specification of who has authorized access to the product (both functionality and data), under what circumstances that access is granted, and to which parts of the product access is allowed.

Motivation

To understand the expectations for confidentiality aspects of the system.

Examples

Only direct managers can see the personnel records of their staff.

Only holders of current security clearance can enter the building.

Fit Criterion

System function name or system data name.

User roles and/or names of people who have clearance.

Considerations

Is there any data that management considers to be sensitive? Is there any data that low-level users do not want management to have access to? Are there any processes that might cause damage or might be used for personal gain? Are there any people who should not have access to the system?

Avoid stating how you will design a solution to the security requirements. For instance, don't "design a password system." Your aim here is to identify the security requirement; the design will then come from this description.

Consider asking for help. Computer security is a highly specialized field, and one where improperly qualified people have no business. If your product has need of more than average security, we advise you to make use of a security consultant. Such consultants are not cheap, but the results of inadequate security can be even more expensive.

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

Acceptance Tests: List ID# and/or names here . . .

15b Integrity Requirements

SV: These requirements address the protection of data(bases) from intentional or accidental corruption, loss, or theft.

Content

Specification of the required integrity of databases and other files, and of the product itself.

Motivation

To understand the expectations for the integrity of the product's data. To specify what the product will do to ensure its integrity in the case of an unwanted happening such as attack from the outside or unintentional misuse by an authorized user.

Examples

The product shall prevent incorrect data from being introduced.

The product shall protect itself from intentional abuse.

Considerations

Organizations are relying more and more on their stored data. If this data should be come corrupt or incorrect—or disappear—then it could be a fatal blow to the organization. For example, almost half of small businesses go bankrupt after a fire destroys their computer systems. Integrity requirements are aimed at preventing complete loss, as well as corruption, of data and processes.

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

Acceptance Tests: List ID# and/or names here . . .

15c Privacy Requirements

SV: These requirements address data that must remain confidential, such as medical records or other personally identifiable data. Laws often apply. (See also section 20.)

Content

Specification of what the product has to do to ensure the privacy of individuals about whom it stores information. The product must also ensure that all laws related to privacy of an individual's data are observed.

Motivation

To ensure that the product complies with the law, and to protect the individual privacy of your customers. Few people today look kindly on organizations that do not observe their privacy.

Examples

The product shall make its users aware of its information practices before collecting data from them.

The product shall notify customers of changes to its information policy.

The product shall reveal private information only in compliance with the organization's information policy.

The product shall protect private information in accordance with the relevant privacy laws and the organization's information policy.

Considerations

Privacy issues may well have legal implications, and you are advised to consult with your organization's legal department about the requirements to be written in this section.

Consider what notices you must issue to your customers before collecting their personal information. A notice might go so far as to warn customers that you intend to put a cookie in their computer. Also, do you have to do anything to keep customers aware that you hold their personal information?

Customers must always be in a position to give or withhold consent when their private data is collected or stored. Similarly, customers should be able to view any private data and, where appropriate, ask for correction of the data.

Also consider the integrity and security of private data—for example, when you are storing credit card information.

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

Acceptance Tests: List ID# and/or names here . . .

15d Audit Requirements

SV: This section applies when a system must provide support for transaction auditing,

such as some financial or medical systems.

Content

Specification of what the product has to do (usually retain records) to permit the required audit checks.

Motivation

To build a system that complies with the appropriate audit rules.

Considerations

This section may have legal implications. You are advised to seek the approval of your organization's auditors regarding what you write here.

You should also consider whether the product should retain information on who has used it. The intention is to provide security such that a user may not later deny having used the product or participated in some form of transaction using the product.

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

Acceptance Tests: List ID# and/or names here . . .

15e Immunity Requirements

SV: This section addresses the system's ability to resist viruses, worms, Trojan Horses, etc.

Content

The requirements for what the product has to do to protect itself from infection by unauthorized or undesirable software programs, such as viruses, worms, and Trojan horses, among others.

Motivation

To build a product that is as secure as possible from malicious interference.

Considerations

Each day brings more malevolence from the unknown, outside world. People buying

software, or any other kind of product, expect that it can protect itself from outside interference.

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

Acceptance Tests: List ID# and/or names here . . .

16 Usability and Humanity Requirements

SV: This section is concerned with requirements that make the product usable and ergonomically acceptable to its hands-on users.

16a Ease of Use Requirements

SV: This section addresses the ease with which the intended audience can use the system properly, and conversely the difficulty with which they can use it improperly.

Content

This section describes your client's aspirations for how easy it is for the intended users of the product to operate it. The product's usability is derived from the abilities of the expected users of the product and the complexity of its functionality.

The usability requirements should cover properties such as these:

- *Efficiency of use: How quickly or accurately the user can use the product.*

- *Ease of remembering: How much the casual user is expected to remember about using the product.*

- *Error rates: For some products it is crucial that the user commits very few, or no, errors.*

- *Overall satisfaction in using the product: This is especially important for commercial, interactive products that face a lot of competition. Web sites are a good example.*

- *Feedback: How much feedback the user needs to feel confident that the product is actually accurately doing what the user expects. The necessary degree of feedback will be higher for some products (e.g., safety-critical products) than for others.*

Motivation

To guide the product's designers toward building a product that meets the expectations of its eventual users.

Examples

The product shall be easy for 11-year-old children to use.

The product shall help the user to avoid making mistakes.

The product shall make the users want to use it.

The product shall be used by people with no training, and possibly no understanding of English.

Fit Criterion

These examples may seem simplistic, but they do express the intention of the client. To completely specify what is meant by the requirement, you must add a measurement against which it can be tested—that is, a fit criterion. Here are the fit criteria for the preceding examples:

Eighty percent of a test panel of 11-year-old children shall be able to successfully complete [list of tasks] within [specified time].

One month's use of the product shall result in a total error rate of less than 1 percent.

An anonymous survey shall show that 75 percent of the intended users are regularly using the product after a three-week familiarization period.

Considerations

Refer to section 3, Users of the Product, to ensure that you have considered the usability requirements from the perspective of all the different types of users.

It may be necessary to have special consulting sessions with your users and your client to determine whether any special usability considerations must be built into the product.

You could also consider consulting a usability laboratory experienced in testing the usability of products that have a project situation (sections 1–7 of this template) similar to yours.

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

Acceptance Tests: List ID# and/or names here . . .

16b Personalization and Internationalization Requirements

SV: This section addresses the ease with which the system can be configured for personal preferences, and for things such as language, currency, units, symbols, etc.

Content

This section describes the way in which the product can be altered or configured to take into account the user's personal preferences or choice of language.

The personalization requirements should cover issues such as the following:

- *Languages, spelling preferences, and language idioms*
- *Currencies, including the symbols and decimal conventions*
- *Personal configuration options*

Motivation

To ensure that the product's users do not have to struggle with, or meekly accept, the builder's cultural conventions.

Examples

The product shall retain the buyer's buying preferences.

The product shall allow the user to select a chosen language.

Considerations

Consider the country and culture of the potential customers and users of your product. Any out-of-country users will welcome the opportunity to convert to their home spelling and expressions.

By allowing users to customize the way in which they use the product, you give them the opportunity to participate more closely with your organization as well as enjoy their own personal user experience.

You might also consider the configurability of the product. Configurability allows

different users to have different functional variations of the product.

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

Acceptance Tests: List ID# and/or names here . . .

16c Learning Requirements

SV: Requirements related to how easy it is for the intended audience to learn to use the product.

Content

Requirements specifying how easy it should be to learn to use the product. This learning curve ranges from zero time for products intended for placement in the public domain (e.g., a parking meter or a web site) to a considerable amount of time for complex, highly technical products. (We know of one product where it was necessary for graduate engineers to spend 18 months in a training program before being qualified to use the product.)

Motivation

To quantify the amount of time that your client feels is allowable before a user can successfully use the product. This requirement guides designers to understand how users will learn the product. For example, designers may build elaborate interactive help facilities into the product, or the product may be packaged with a tutorial. Alternatively, the product may have to be constructed so that all of its functionality is apparent upon first encountering it.

Examples

The product shall be easy for an engineer to learn.

A clerk shall be able to be productive within a short time.

The product shall be able to be used by members of the public who will receive no training before using it.

The product shall be used by engineers who will attend five weeks of training before using the product.

Fit Criterion

An engineer shall produce a [specified result] within [specified time] of beginning to use the product, without needing to use the manual.

After receiving [number of hours] training a clerk shall be able to produce [quantity of specified outputs] per [unit of time].

[Agreed percentage] of a test panel shall successfully complete [specified task] within [specified time limit].

The engineers shall achieve [agreed percentage] pass rate from the final examination of the training.

Considerations

Refer to section 3, Users of the Product, to ensure that you have considered the ease of learning requirements from the perspective of all the different types of users.

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

Acceptance Tests: List ID# and/or names here . . .

16d Understandability and Politeness Requirements

SV: These requirements relate to how intuitively the intended audience understands what the program does, what its messages mean, and how to use it. Definitely related to ease of use, (section 16a), but more specifically addressing comprehension of the program output, instructions, and other messages.

This section is concerned with discovering requirements related to concepts and metaphors that are familiar to the intended end users.

Content

This specifies the requirement for the product to be understood by its users. While "usability" refers to ease of use, efficiency, and similar characteristics, "understandability" determines whether the users instinctively know what the product will do for them and how it fits into their view of the world. You can think of understandability as the product being polite to its users and not expecting them to

know or learn things that have nothing to do with their business problem.

Motivation

To avoid forcing users to learn terms and concepts that are part of the product's internal construction and are not relevant to the users' world. To make the product more comprehensible and thus more likely to be adopted by its intended users.

Examples

The product shall use symbols and words that are naturally understandable by the user community.

The product shall hide the details of its construction from the user.

Considerations

Refer to section 3, Users of the Product, and consider the world from the point of view of each of the different types of users.

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

Acceptance Tests: List ID# and/or names here . . .

16e Accessibility Requirements

SV: Requirements related to use of the product by individuals with disabilities.

Content

The requirements for how easy it should be for people with common disabilities to access the product. These disabilities might be related to physical disability or visual, hearing, cognitive, or other abilities.

Motivation

In many countries it is required that some products be made available to the disabled. In any event, it is self-defeating to exclude this sizable community of potential customers.

Examples

The product shall be usable by partially sighted users.

The product shall conform to the Americans with Disabilities Act.

Considerations

Some users have disabilities other than the commonly described ones. In addition, some partial disabilities are fairly common. A simple, and not very consequential, example is that approximately 20 percent of males are red-green colorblind.

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

Acceptance Tests: List ID# and/or names here . . .

16f User Documentation Requirements

SV: List of the user documentation to be supplied as part of the product.

Content

List of the user documentation to be supplied as part of the product.

Motivation

To set expectations for the documentation and to identify who will be responsible for creating it.

Examples

Technical specifications to accompany the product.

User manuals.

Service manuals (if not covered by the technical specification).

Emergency procedure manuals (e.g., the card found in airplanes).

Installation manuals.

Considerations

Which documents do you need to deliver, and to whom? Bear in mind that the answer to this questions depends on your organizational procedures and roles.

For each document, consider these issues:

- *The purpose of the document*
- *The people who will use the document*
- *Maintenance of the document*

What level of documentation is expected? Will the users be involved in the production of the documentation? Who will be responsible for keeping the documentation up-to date? What form will the documentation take?

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

Acceptance Tests: List ID# and/or names here . . .

16g Training Requirements

SV: A description of the training needed by users of the product.

Content

A description of the training needed by users of the product.

Motivation

To set expectations for the training. To identify who is responsible for creating and providing that training.

Considerations

What training will be necessary? Who will design the training? Who will provide the training?

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

Acceptance Tests: List ID# and/or names here . . .

17 Look and Feel Requirements

17a Appearance Requirements

SV: These requirements address things such as the colors, fonts, and logos used, often to reflect corporate branding or similarity to related products. See also style in the next section.

Content

The section contains requirements relating to the spirit of the product. Your client may have made particular demands for the product, such as corporate branding, colors to be used, and so on. This section captures the requirements for the appearance. Do not attempt to design it until the appearance requirements are known.

Motivation

To ensure that the appearance of the product conforms to the organization's expectations.

Examples

The product shall be attractive to a teenage audience.

The product shall comply with corporate branding standards.

Fit Criterion

A sampling of representative teenagers shall, without prompting or enticement, start using the product within four minutes of their first encounter with it.

The office of branding shall certify the product complies with the current standards.

Considerations

Even if you are using prototypes, it is important to understand the requirements for the appearance. The prototype is used to help elicit requirements; it should not be thought of as a substitute for the requirements.

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

Acceptance Tests: List ID# and/or names here . . .

17b Style Requirements

SV: Style requirements address the impression the product makes upon users, such as professionalism for a tax accounting package, friendliness for a children's game, or how "cool" it is for a teenage audience. Product packaging may also be addressed here, and/or appearance in the previous section.

Content

Requirements that specify the mood, style, or feeling of the product, which influences the way a potential customer will see the product. Also, the stakeholders' intentions for the amount of interaction the user is to have with the product.

In this section, you would also describe the appearance of the package if this is to be a manufactured product. The package may have some requirements as to its size, style, and consistency with other packages put out by your organization. Keep in mind the European laws on packaging, which require that the package not be significantly larger than the product it encloses.

The style requirements that you record here will guide the designers to create a product as envisioned by your client.

Motivation

Given the state of today's market and people's expectations, we cannot afford to build products that have the wrong style. Once the functional requirements are satisfied, it is often the appearance and style of products that determine whether they are successful. Your task in this section is to determine precisely how the product shall appear to its intended consumer.

Example

The product shall appear authoritative.

Fit Criterion

After their first encounter with the product, 70 percent of representative potential customers shall agree they feel they can trust the product.

Considerations

The look and feel requirements specify your client's vision of the product's appearance. The requirements may at first seem to be rather vague (e.g., "conservative and

professional appearance"), but these will be quantified by their fit criteria. The fit criteria give you the opportunity to extract from your client precisely what is meant, and give the designer precise instructions on what he is to accomplish.

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

Acceptance Tests: List ID# and/or names here . . .

18 Operational and Environmental Requirements

18a Expected Physical Environment

SV: These requirements relate to the physical environment in which the product will operate.

Content

This section specifies the physical environment in which the product will operate.

Motivation

To highlight conditions that might need special requirements, preparations, or training. These requirements ensure that the product is fit to be used in its intended environment.

Examples

The product shall be used by a worker, standing up, outside in cold, rainy conditions.

The product shall be used in noisy conditions with a lot of dust.

The product shall be able to fit in a pocket or purse.

The product shall be usable in dim light.

The product shall not be louder than the existing noise level in the

environment. Considerations

The work environment: Is the product to operate in some unusual environment? Does this lead to special requirements? Also see section 11, Usability and Humanity Requirements.

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

Acceptance Tests: List ID# and/or names here . . .

18b Requirements for Interfacing with Adjacent Systems

SV: This section describes the requirements to interface with partner applications and/or devices that the product needs to successfully operate.

Content

This section describes the requirements to interface with partner applications and/or devices that the product needs to successfully operate.

Motivation

Requirements for the interfaces to other applications often remain undiscovered until implementation time. Avoid a high degree of rework by discovering these requirements early.

Examples

The products shall work on the last four releases of the five most popular browsers.

The new version of the spreadsheet must be able to access data from the previous two versions.

Our product must interface with the applications that run on the remote weather stations.

Fit Criterion

For each inter-application interface, specify the following

elements: ● The data content

● The physical material content

● The medium that carries the interface

● The frequency

● The volume

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

Acceptance Tests: List ID# and/or names here . . .

18c Productization Requirements

SV: Requirements related to the distribution and/or installation of the product.

Content

Any requirements that are necessary to make the product into a distributable or salable item. It is also appropriate to describe here the operations needed to install a software product successfully.

Motivation

To ensure that if work must be done to get the product out the door, then that work becomes part of the requirements. Also, to quantify the client's and users' expectations about the amount of time, money, and resources they will need to allocate to install the product.

Examples

The product shall be distributed as a ZIP file.

The product shall be able to be installed by an untrained user without recourse to separately printed instructions.

The product shall be of a size such that it can fit on one CD.

Considerations

Some products have special needs to turn them into a salable or usable product. You might consider that the product has to be protected such that only paid-up customers can access it.

Ask questions of your marketing department to discover unstated assumptions that have been made about the specified environment and the customers' expectations of how long installation will take and how much it will cost.

Most commercial products have some needs in this area.

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

Acceptance Tests: List ID# and/or names here . . .

18d Release Requirements

SV: Specification of the intended release cycle for the product and the form that the release shall take.

Content

Specification of the intended release cycle for the product and the form that the release shall take.

Motivation

To make everyone aware of how often you intend to produce new releases of the product.

Examples

The maintenance releases will be offered to end users once a year.

Each release shall not cause previous features to fail.

Fit Criterion

Description of the type of maintenance plus the amount of effort budgeted for it. Considerations

Do you have any existing contractual commitments or maintenance agreements that might be affected by the new product?

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

Acceptance Tests: List ID# and/or names here . . .

19 Cultural and Political Requirements

19a Cultural Requirements

SV: This section contains requirements that are specific to the sociological factors that affect the acceptability of the product. If you are developing a product for foreign markets, then these requirements are particularly relevant. Bear in mind that "cultural groups" may also apply to population subgroups such as teenagers, the elderly, or ironworkers.

Content

This section contains requirements that are specific to the sociological factors that affect the acceptability of the product. If you are developing a product for foreign markets, then these requirements are particularly relevant.

Motivation

To bring out in the open requirements that are difficult to discover because they are outside the cultural experience of the developers.

Examples

The product shall not be offensive to religious or ethnic groups.

The product shall be able to distinguish between French, Italian, and British road numbering systems.

The product shall keep a record of public holidays for all countries in the European Union and for all states in the United States.

Considerations

Question whether the product is intended for a culture other than the one with which you are familiar. Ask whether people in other countries or in other types of organizations will use the product. Do these people have different habits, holidays, superstitions, or cultural norms that do not apply to your own culture? Are there colors, icons, or words that have different meanings in another cultural environment?

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

Acceptance Tests: List ID# and/or names here . . .

19b Political Requirements

SV: Requirements included strictly to make "the boss" happy, either internally to the development company, or internally to the client company, or possibly an external third party.

Content

This section contains requirements that are specific to the political factors that affect the acceptability of the product.

Motivation

To understand requirements that sometimes appear irrational.

Examples

The product shall be installed using only American-made

components. The product shall make all functionality available to the

CEO.

Considerations

Did you intend to develop the product on a Macintosh, when the office manager has laid down an edict that only Windows machines are permitted?

Is a director also on the board of a company that manufactures products similar to the one that you intend to build?

Whether you agree with these political requirements has little bearing on the outcome. The reality is that the system has to comply with political requirements even if you can find a better, more efficient, or more economical solution. A few probing questions here may save some heartache later.

The political requirements might be purely concerned with the politics inside your organization. However, in other situations you may need to consider the politics inside your customers' organizations or the national politics of the country.

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

Acceptance Tests: List ID# and/or names here . . .

20 Legal Requirements

20a Compliance Requirements

SV: A statement specifying the legal requirements for this system, often referring to relevant laws and/or requiring approval by the legal department.

Content

A statement specifying the legal requirements for this system.

Motivation

To comply with the law so as to avoid later delays, lawsuits, and legal

fees. Examples

Personal information shall be implemented so as to comply with the Data Protection Act.

Fit Criterion

Lawyers' opinion that the product does not break any laws.

Considerations

Consider consulting lawyers to help identify the legal requirements.

Are there any copyrights or other intellectual property that must be protected? Conversely, do any competitors have copyrights on which you might be in danger of infringing?

Is it a requirement that developers have not seen competitors' code or even have worked for competitors?

The Sarbanes-Oxley (SOX) Act, the Health Insurance Portability and Accountability Act (HIPAA) and the Gramm-Leach-Bliley Act may have implications for you. Check with your company lawyer.

Might any pending legislation affect the development of this system?

Are there any aspects of criminal law you should consider?

Have you considered the tax laws that affect your product?

Are there any labor laws (e.g., working hours) relevant to your product?

ID# - Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

Acceptance Tests: List ID# and/or names here . . .

20b Standards Requirements

SV: These requirements specify documented standards to which the product must conform, as opposed to legal regulations.

Content

A statement specifying applicable standards and referencing detailed standards descriptions. This does not refer to the law of the land—think of it as an internal law imposed by your company.

Motivation

To comply with standards so as to avoid later delays.

Example

The product shall comply with MilSpec standards.

The product shall comply with insurance industry standards.

The product shall be developed according to SSADM standard development steps.

Fit Criterion

The appropriate standard-keeper certifies that the standard has been adhered to.

Considerations

It is not always apparent that there are applicable standards because their existence is often taken for granted. Consider the following:

- *Do any industry bodies have applicable standards?*
- *Does the industry have a code of practice, watchdog, or ombudsman?*
- *Are there any special development steps for this type of product?* **ID# -**

Name

Description: Your description here . . .

Rationale: Your rationale here . . .

Fit Criterion: Your fit criteria here . . .

Acceptance Tests: List ID# and/or names here . . .

21 Requirements Acceptance Tests

SV: Every requirement must have one or more acceptance tests associated with it, to confirm that the requirement has been met. At this point these tests are not yet completely specified – A one- or two-sentence description of each test will suffice. Note that some tests may verify more than one requirement, and that some requirements may require multiple tests for their confirmation.

21a Requirements – Test Correspondence Summary

SV: The following sample table is available from the CS 440 web site as "Sample Requirement Test Correspondence Table.xlsx" It is recommended that you work with the table in Excel, and then drag it into the document when it is completed. Depending on the number of requirements and/or tests included, it may be necessary to use

multiple tables, and/or use landscape mode. Every row and every column of the table should include at least one X. Below the table list the ID #, name, and short description of each individual acceptance test.

Requirements

Test

Test 1 X
Test 2 X X
Test 3 X X
Test 4 X X
Test 5
Test 6
Test 7
Test 8
Test 9
Test 10
Test 11
Test 12
Test 13
Test 14
Test 15

Table 1 - Requirements - Acceptance Tests Correspondence

21b Acceptance Test Descriptions

SV: Provide a brief description of each acceptance test. Detailed test specifications will appear in a separate document, which may be referenced here when available.

ID # - Name

Description: Your description here . . .

III Design

22 Design Goals

SV: Identify the important design goals that are to be optimized in the proposed design. Content

Design goals are important properties of the system to be optimized, and which may affect the overall design of the system. For example computer games place a higher priority on speed than accuracy, and so the physics engine for a computer game may make some rough approximations and assumptions that allow it to run as fast as possible while sacrificing accuracy, whereas the physics calculations performed by NASA must be much more rigorously correct, even at the expense of speed.

Note an important difference between design goals and requirements: Requirements include specific values that must be met in order for the product to be acceptable to the client, whereas design goals are properties that the designers strive to make "as good as possible", without specific criteria for acceptability. (Note also that the same property may appear in both a requirement and a design goal, so a design goal may be to make the system run as fast as possible, with a requirement that says any speed below a certain specified threshold is unacceptable.)

Your text goes here . . .

23 Current System Design

*SV: **IF** the proposed new system is to replace an existing system, then the current system should be described here. Otherwise insert a brief statement that there is no pre-existing system.*

Your text goes here . . .

24 Proposed System Design

This section will make heavy use of class diagrams, and also sequence and deployment diagrams where noted. However don't overlook finite state, activity, communication, or other diagram types as needed for effective communication.

24a Initial System Analysis and Class Identification

SV: Perform grammatical and similar analyses to identify the most import and obviously needed classes, and to organize them into an initial class structure. An initial class diagram is appropriate, containing few if any internal details.

Your text goes here . . .

24b Dynamic Modelling of Use-Cases

SV: Insert sequence diagrams of (at least the most important) use-cases, as a means of identifying other needed classes.

Content

Include sequence diagrams of each important use-case here. This is a first step towards identifying preliminary objects. (If the sequence diagram would be too big to fit, then it can either be broken down into pieces or a communication diagram can be used in its place.)

Your text goes here . . .

24c Proposed System Architecture

SV: Identify the Software Architecture to be applied to this project, such as Client Server, Repository, MVC, etc., along with justification for the choice.

Your text goes here . . .

24d Initial Subsystem Decomposition

SV: A slightly more detailed class diagram, showing the classes identified in sections 24a, 24b, and 0 above, partitioned into subsystems. For each subsystem provide a brief description of the subsystem, including its key responsibilities. There should still be few if any internal details.

Your text goes here . . .

25 Additional Design Considerations

SV: The sections listed here do not need to be presented in the order given, and may not all be relevant for any particular project. Those that are relevant can help identify additional classes that are needed as a result.

25a Hardware / Software Mapping

SV: This is particularly important for distributed systems, such as those employing a client-server architecture. Use a deployment diagram to indicate which subsystems are mapped onto which piece(s) of hardware, and what communication subsystems need to be added to the system as a result.

Your text goes here . . .

SV: The communication subsystem required for this program is a data retrieval system that constantly grabs data from the DIVVY API since it gets updated in real time.

25b Persistent Data Management

SV: Document the classes and perhaps subsystems necessary to store persistent data when the system shuts down, and to restore that data when the system starts back up again.

Reiterate key data structures and information as necessary for the understanding of this design phase. Refer the reader back to the data dictionary in section I7c above to avoid undue repetition, while reviewing only the most relevant items here.

Your text goes here . . .

The classes and data structures for storing location data from the API will be in the form of a set or a graph. The data will be loaded once at startup and then will update manually as the application continues to run.

The graph, set, and map data structures are useful since data retrieval is faster and data relevance is stored accordingly.

25c Access Control and Security

SV: Identify the access control and security concerns for this system, and the new classes and/or subsystems that must be added to handle those concerns.

Your text goes here . . .

To access the application, each user must login with their registered credentials and their data will be stored and saved in the system.

25d Global Software Control

SV: Identify the global software control concerns for this system, and the new classes and/or subsystems that must be added to handle those concerns.

Your text goes here . . .

The global software control concerns for this system would be controlling traffic of multiple users. Classes or subsystems to be added could be to link client/users to a shared server and allow the server to be the centralized system for handling traffic and client requests.

25e Boundary Conditions

SV: Identify the boundary condition concerns for this system, and the new classes and/or subsystems that must be added to handle those concerns. In particular consider startup, shutdown (normal or abnormal), and the creation and/or maintenance of any configuration files, databases, or similar supporting data files.

Your text goes here . . .

Boundary condition concerns for this application would include creating a server to handle and manage multiple users. This would prevent issues regarding users attempting to search and create new menu items from scratch rather than saving their data. Storing the users prior data into a server will allow the application to suggest other items of similar interest to the based on data already collected.

25f User Interface

SV: Include a preliminary user interface design here, possibly as a rough sketch or other mockup, in order to identify additional classes needed to implement the interface.

The final user interface design will normally be developed by appropriate experts in that area. However it is appropriate to include an initial design here, including possibly a low- or high- fidelity sketch/mockup, in order to identify key classes necessary to implement the user interface, such as forms and dialog windows. It may also go towards addressing usability and/or look-and-feel requirements, and/or identifying other overlooked components.

Your text goes here . . .

25g Application of Design Patterns

SV: Any design patterns applied as a result of previous sections should have been addressed there, and identified as such at the time. Use this section to document only the additional design patterns that were not previously covered elsewhere. (If any.)

Your text goes here . . .

26 Final System Design

SV: Include here the final version of the overall system design, incorporating all the subsystems and classes added as a result of additional design considerations. Multiple diagrams may be needed, possibly starting with an overall package diagram showing all the different subsystems and the (important) classes contained within each one. Still not a lot of internal details.

Your text goes here . . .

27 Object Design

This section documents the internal details of each class, to the extent that they can be designed at this time. Included should be the class interfaces (public method signatures and responsibilities) and constraints. It is probably best to break this section up into subsections corresponding to subsystems as documented above, and/or

by (Java) packages if those are designed. It may also be appropriate to address additional design pattern considerations here, but not to the point of being redundant of previous documentation.

Certain methods, such as simple getters, setters, and constructors are not always documented, unless there is something special about them such as in the Singleton or Factory Method design patterns.

27a Packages

SV: If the design involves assigning classes to packages (.e.g Java packages), then the packages to be created should be documented here.

Your text goes here . . .

27b Subsystem I

Your text goes here . . .

27c Subsystem II

Your text goes here . . .

27d etc.

Your text goes here . . .

IV Project Issues

28 Open Issues

SV: Issues that have been raised and do not yet have a conclusion.

Content

A statement of factors that are uncertain and might make significant difference to the product.

Motivation

To bring uncertainty out in the open and provide objective input to risk

analysis. Examples

Our investigation into whether the new version of the processor will be suitable for our application is not yet complete.

The government is planning to change the rules about who is responsible for gritting the motorways, but we do not know what those changes might be.

Considerations

Are there any issues that have come up from the requirements gathering that have not yet been resolved? Have you heard of any changes that might occur in the other organizations or systems on your context diagram? Are there any legislative changes that might affect your system? Are there any rumors about your hardware or software suppliers that might have an impact?

Your text goes here . . .

The application is currently not developed for mobile devices.

Content

Some unknown factors include runtime of certain algorithms, and memory.

Motivation

To highlight some important issues to be the focus for continuing this project.

Examples

Mobile development is large in two areas, IOS and Android. Figuring out which one is most suitable for our specific application is something we are still considering.

A plan to develop in both IOS and Android is also a discussion, but would still raise problems with workload management.

Considerations

More advanced and updated IOS and Android versions may be available in the near future that may make it more complicated to transition into a mobile platform. Planning on which version is available to us currently and what updates future versions will have is another event we are considering.

29 Off-the-Shelf Solutions

SV: Discussion of products or components currently available that could either be incorporated into the new solution or simply used instead of developing (parts of) the new solution. The distinction between sections 35 a, b, and c is subtle, and not very important.

Your text goes here . . .

An alternative solution for making the application more accessible to mobile devices would involve building up the same application from scratch. This is due to the fact that the current GUI version of the application is not compatible with mobile devices and could not be

carried over as of now. However, a new GUI would not require as much development time and work as the backend.