

# **4TB6: SRS**

**Stonecap Solutions - Smart Serve**

Max Turek *turekm*  
Ryan Were *werer*  
Sam Nusselder *nusselds*  
Peter Minbashian *minbashp*  
David Bednar *bednad1*

10/05/2022

# Contents

<b>1</b>	<b>Introduction</b>	<b>5</b>
1.1	Purpose	5
1.2	Definitions, Terms, Acronyms and Abbreviations	5
1.3	Scope	5
1.4	Context Diagram	7
1.5	Monitored and Controlled Variables	7
1.5.1	Monitored Variables	7
1.5.2	Controlled Variables	8
1.5.3	System Responses	8
1.6	Constants	9
1.7	Behaviour Overview	9
1.8	Stakeholders	9
<b>2</b>	<b>Project Overview</b>	<b>10</b>
2.1	Normal Operation	10
2.2	Undesired Event Handling	10
2.3	User Characteristics	10
2.4	Constraints	10
2.5	Assumptions and Dependencies	11
<b>3</b>	<b>Function Requirements</b>	<b>11</b>
3.1	Ordering Drink	11
3.2	Making Drink	12
<b>4</b>	<b>Non-Functional Requirements</b>	<b>13</b>
4.1	Quality requirements of the entire system	13
4.2	Look and Feel Requirements	13
4.3	Usability and Humanity Requirements	13
4.3.1	Ease of use	13
4.3.2	Ease of learning	13
4.4	Performance Requirements	13
4.4.1	Speed requirement	13
4.4.2	Safety critical requirement	14
4.4.3	Precision requirement	14
4.4.4	Reliability and availability requirement	14
4.4.5	Capacity requirement	14
4.5	Operational and Environmental Requirements	14
4.5.1	Expected physical environment	14
4.5.2	Expected technological environment	14
4.5.3	Partner applications	14
4.6	Maintainability and Support Requirements	15
4.6.1	Maintainability	15
4.6.2	Special Maintenance Conditions	15
4.6.3	Portability	15
4.7	Security Requirements	15
4.8	Cultural and Political Requirements	15
4.9	Legal Requirements	15
4.10	Health and Safety Requirements	16

<b>5</b>	<b>Likely Changes to Requirements</b>	<b>16</b>
5.1	Machine Improvements . . . . .	16
5.2	Data Analytics . . . . .	16
<b>6</b>	<b>Unlikely Changes to Requirements</b>	<b>16</b>
<b>7</b>	<b>References</b>	<b>18</b>
<b>8</b>	<b>Appendixes</b>	<b>19</b>
8.1	Reflection . . . . .	19

Table 1: Revision History

<b>Date</b>	<b>Developer(s)</b>	<b>Change</b>
10/05/22	Max Turek	Initial Draft
	Sam Nusselder	
	Ryan Were	
	Peter Minbasian	
11/02/22	David Bednar	Added new functional requirements ODR14 - ODR19 from Hazard Analysis Doc
	Sam Nusselder	

# 1 Introduction

## 1.1 Purpose

Stonecap Solutions aims to solve existing issues in the billion dollar bartending industry relating to the process of fulfilling drink orders. Bars are often busy with many orders being processed through a mental queue by bartenders. This can result in long wait times. Furthermore, many restaurants and bars are susceptible to being understaffed, further exasperating this issue. Cocktails and other drinks are imprecisely made, varying in volume and consistency. When bartenders are rushing around to make these drinks, the risk of spilling them arises. If these issues are severe enough, it could result in unsatisfied customers, decreasing business profits.

Stonecap Solutions aims to solve this problem by developing a drink delivery system that streamlines the process of a customer ordering a drink up to them receiving it. The system would automate the tasks of taking customer orders, making the drinks, and alerting the end user when the drinks are ready. This would result in a system that creates consistent, accurate and timely drinks while avoiding unnecessary spillage.

## 1.2 Definitions, Terms, Acronyms and Abbreviations

Term	Definition
Drink	Drink will refer to any mixed drink or cocktail made up of many or one ingredient.
Web App	Web application that is used by the customer to order drinks, and used by administrators to monitor the device.
Glass	Container that will hold the liquids/ingredients of a drink.
Operator	Usually a bartender or manager that is responsible for maintaining the system and has administrative privileges.
User	Individual that is interacting with the UI and requesting a drink from the system

Table 2: Table of Definitions, Terms, Acronyms and Abbreviations

## 1.3 Scope

The goals of the system are found in Table 3. These goals have determined the following primary behaviours that make up the scope of the project:

1. Ordering Drink: Users are able to interact with a user-interface to order a drink
2. Creating Drink: Drink is accurately and quickly created
3. Serving Drink: Customer is notified drink is ready

Goal	Explanation
Ease of Communication	End users are able to effectively communicate with Smart Serve. Users understand the status of Smart Serve. This would be measured by all customer orders being fulfilled correctly and all users being informed on their order status.
Ease of Use	System is simple and easy to use for all customers. This would be measured by at least 75 percent of users find the product intuitive to use.
Autonomy	Smart Serve is fully autonomous once starting up. This allows users to interact with Smart Serve without any human interaction.
Consistency	Smart Serve continues to pour a consistent drink after 100 pours.
Accuracy	Smart Serve pours drinks with accurate proportions. This would be measured by checking if the individual ingredients of a drink follow the appropriate proportion (i.e. 100 millilitres of rum and 300 millilitres of coke).
Fast Service	Drinks are served to users quickly. One drink is served within 45 seconds.
Operator	The operator had access to the operator controls in the web app. These controls allow the operator to manage ingredients in the web app.

Table 3: Project Goals

The stretch goals shown in Table 4 were not included in the scope because they are goals that will only be implemented once all the primary project goals are implemented.

Stretch Goal	Explanation
Self Cleaning	Smart Serve is able to clean all necessary components without human interaction.
Fool Proof	Any risks posed by realistic system malfunctions should be minimized through smart design.
Smart Recommendations	Smart Serve can provide drink recommendations. This can be measured by drink recommendations being purchased at least 20 percent of the time.

Table 4: Stretch Goals

## 1.4 Context Diagram

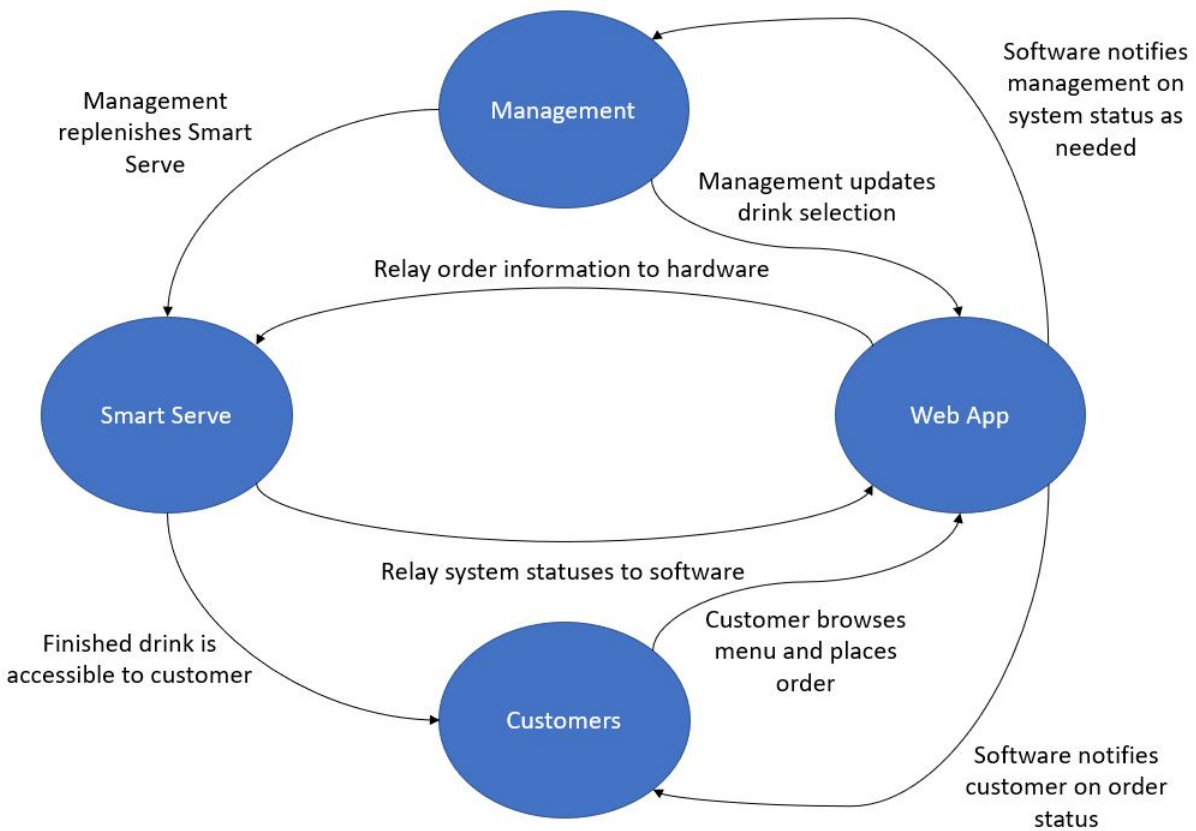


Figure 1: Context Diagram

## 1.5 Monitored and Controlled Variables

### 1.5.1 Monitored Variables

- M.ingredientReady
- M.cupPresent
- M.cupEmpty
- M.username
- M.password
- M.alcoholID
- M.mixID
- M.drinkVolume
- M.orderPlaced

### 1.5.2 Controlled Variables

- C\_valveOpen
- C\_notification

### 1.5.3 System Responses

	<b>C_valveOpen</b>	<b>C_notification</b>
no stimulus	e_off	e_off
!M_ingredientReady	e_off	e_on
!M_cupPresent	e_off	e_on
!M_username	e_off	e_on
!M_password	e_off	e_on
!M_alcoholID	e_off	e_on
!M_mixID	e_off	e_on
!M_drinkVolume	e_off	e_on
!M_orderPlaced	e_off	e_off
M_ingredientReady . M_cupPresent . M_username . M_password . M_alcoholID . M_mixID . M_drinkVolume . M_orderPlaced . M_cupEmpty	e_on	e_off
!M_cupEmpty	e_off	e_off

Figure 2: System Response Chart



	State
no stimulus	Device in Standby
!M_ingredientReady	Notification to Operator
!M_cupPresent	Notification to Operator
!M_username	Notification to User or Operator
!M_password	Notification to User or Operator
!M_alcoholID	Notification to User
!M_mixID	Notification to User
!M_drinkVolume	Notification to User
!M_orderPlaced	Notification to User
M_ingredientReady . M_cupPresent . M_username . M_password . M_alcoholID . M_mixID . M_drinkVolume . M_orderPlaced	Mix Drink
!M_cupEmpty	Notification to User

Figure 3: Stimulus Effect Chart

## 1.6 Constants

- Ingredient flow rate
- Cup size
- Location of cup
- Ingredient vat volume

## 1.7 Behaviour Overview

Smart Serve is an autonomous system once the user orders a drink. The user can interact with the web app to place an order with Smart Serve. From here the system is entirely autonomous. Once the user is at the front of the queue, the drink will be created and then served where it can be picked up. Smart Serve will remain in standby mode when not in use, and wait for instructions.

## 1.8 Stakeholders

- Stonecap Solutions
- Restaurants and business owners that serve drinks
- Consumers seeking an autonomous drink creation system

- Bartenders
- Customers at bars

## 2 Project Overview

### 2.1 Normal Operation

Smart Serve is intended for users to order drinks and have their drinks robotically made for them. Users will be able to interact with Smart Serve using their phone allowing multiple users to simultaneously order drinks. This will also allow users to socialize with friends while ordering drinks as opposed to waiting in line. Once the order is processed, Smart Serve will begin creating the drink. This process will be completely autonomous. Once the drink is complete, the user will be notified. This will again save the user time waiting in lines for their drink to be made.

### 2.2 Undesired Event Handling

In the case that an undesired event occurs, Smart Serve should return to a safe state. During the transition into a safe state, Smart Serve should ensure that no harm is caused to the user or itself. This would be done by pausing the current action, notifying the operator to ensure no one is physically interacting with Smart Serve and allowing Smart Serve to return to the home state. The operator would then need to reset Smart Serve and send an alert to Smart Serve to notify that Smart Serve will restart the previous action. This would ensure that the user would still get their drink created at the appropriate spot in queue, while also ensuring the safety of the users and Smart Serve.

### 2.3 User Characteristics

There are two intended users for Smart Serve: Managers and Customers.

Managers are grouped as people who intended on using the system to replace bartenders. They own/operate an establishment that sells alcohol. The establishment is busy enough that automating these tasks is warranted to improve service time and reduce labor costs.

Customers would be classified as groups of people who frequently attend establishments that serve alcohol. These customers drink alcohol themselves, most likely drinking cocktails or mixed drinks. They own smartphones and have a basic understanding of modern technology and can navigate through a basic web app.

### 2.4 Constraints

Our product can only function within its given constraints. Some limitations include but not limited to:

- Reliability of internet service via data or WiFi for UI and Cloud Operations
- Liquor laws and regulations involving the service of alcohol
- Sanitary regulations pertaining to the service of drinks

## 2.5 Assumptions and Dependencies

The following assumptions have been made while creating Smart Serve:

- All users are of legal drinking age
- All users are in a safe health condition to consume alcohol, including intoxication level
- All users have a tab created that is assumed to be paid in full once they have finished ordering all their drinks
- Smart Serve will be used in an indoor environment
- Ingredient flow rate is kept constant
- All cup sizes will be identical
- All cups will be plastic
- Smart Serve will be kept clean and not restricted due to usage
- Smart Serve will always have an internet connection
- Smart Serve will always have electrical power
- Ingredient viscosity is consistent

## 3 Function Requirements

### 3.1 Ordering Drink

**ODR2:** User can scan QR code launching smart-serve web app

**ODR3:** User can access the menu and select drinks to order

**ODR4:** User can cancel or change order while waiting for drink to be made

**ODR5:** User can view place/time remaining for the drink to be made

**ODR6:** User is notified if drink ingredients are out of inventory

**ODR7:** User is notified when drink is done

**ODR8:** Web app gives an intoxication test when the customer has ordered too many drinks too quickly

**Rationale:** This is done to comply with Ontario Laws to ensure no customer is served when already intoxicated

**ODR9:** Operator is notified if drink ingredients are out of inventory

**ODR10:** Operator inputs all ingredients available for drinks into the web app

**ODR11:** Operator inputs dispenser location of each ingredient into web app

**Rationale:** This is done to allow for more flexibility with drink mixing and make the system more universal.

**ODR12:** User gets notified when the drink is ready via web app

**ODR13:** Web app keeps track of drinks made

**Rationale:** Data used for analytics to track popular drinks/ingredients or for inventory check

**ODR14:** Server decrypts drink name and gathers ingredients

**ODR15:** User's drink is the same drink they ordered on the app

**ODR16:** User can login successfully into the app

**ODR17:** User has user permissions

**ODR18:** Electrical components of high exposure to liquids have waterproof covering

**ODR19:** Have a robust and durable architecture/components

## 3.2 Making Drink

**MDR1.** Hardware receives ingredients from server

**MDR2.** Hardware dispenses the appropriate amount of liquid from respective bottle

**MDR3.** Hardware can detect if cup is present

**MDR4.** Hardware can detect if cup is empty or full

**MDR5.** Hardware sends a notification to the operator that the drink is done

**MDR6.** Hardware can detect whether an ingredient is out of inventory

**MDR7.** Hardware communicates with the operator on order status (i.e. cup not present, a cup full, out of ingredients, order ready)

**MDR8.** Hardware displays that an order is ready so the user can remove it

**MDR9.** Hardware will not start the next order until the completed drink is removed, and new empty cup is placed in dispensing area

**MDR10.** Hardware does not begin a new drink until the previous drink is removed

**MDR11.** Hardware sends error messaging to the web app (Operator Side) when the system is malfunctioning

**MDR12.** Hardware must complete all drinks in queue if internet access is terminated

**MDR13.** Hardware must restart drink and resume queue if power is lost

## **4 Non-Functional Requirements**

### **4.1 Quality requirements of the entire system**

**QR1.** Smart Serve must be robust

### **4.2 Look and Feel Requirements**

**LFR1.** The web app must be intuitive and visually appealing

**LFR2.** Smart Serve must have all electrical and mechanical components concealed

### **4.3 Usability and Humanity Requirements**

#### **4.3.1 Ease of use**

**UHR1.** Smart Serve must serve the drink at a height that is easy to grab for an average adult

**UHR2.** Smart Serve must make grabbing the drink easy for the user, without blockages or inconveniences

**UHR3.** The web app must have a simple payment system

**UHR4.** The web app must make it easy to navigate and select drinks

#### **4.3.2 Ease of learning**

**UHR5.** Finding and gaining access to the web app must be easy

### **4.4 Performance Requirements**

#### **4.4.1 Speed requirement**

**PR1.** The drink must be ready and served within 45 seconds of when the drink began to be made

**PR2.** The web app must be responsive to user inputs

**PR3.** The web app must add a users order to the queue of orders within 20 seconds of ordering

#### **4.4.2 Safety critical requirement**

**PR4.** Smart Serve must not release more than 1.1x the amount of expected alcohol

**Rationale:** Consuming more alcohol than expected can be dangerous for a user

#### **4.4.3 Precision requirement**

**PR5.** Smart Serve must be able to measure drink ingredients within 10% of the expected value

#### **4.4.4 Reliability and availability requirement**

**PR6.** Smart Serve must be able to create the correct drink every time as long as it has the correct ingredients

#### **4.4.5 Capacity requirement**

**PR7.** Smart Serve must be able to store up to 8 ingredients

**PR8.** Smart Serve must be able to store up to 2 liters of each ingredient

### **4.5 Operational and Environmental Requirements**

#### **4.5.1 Expected physical environment**

**OER1.** Smart Serve must not be exposed to rain or poor weather

**OER2.** Smart Serve must be placed upright on a flat surface

#### **4.5.2 Expected technological environment**

**OER4.** Smart Serve must have access to wifi

**OER5.** Smart Serve must have access to electricity

**OER6.** The user must have internet access and a device to access the web app

#### **4.5.3 Partner applications**

**OER7.** Smart Serve must be equipped with a cup to serve a drink

## **4.6 Maintainability and Support Requirements**

### **4.6.1 Maintainability**

**MSR1.** Smart Serve should be cleaned once every month

**MSR2.** Smart Serve must be restocked with new ingredients to be able to serve drinks

### **4.6.2 Special Maintenance Conditions**

**MSR3.** No electronic or water sensitive components of Smart Serve can get wet during cleaning

**MSR4.** The operator must add the ingredients to the web app when Smart Serve is restocked

### **4.6.3 Portability**

**MSR5.** Smart Serve must be modular and easy to transport

**MSR6.** Smart Serve must be under 100 pounds

## **4.7 Security Requirements**

**SR1.** The users payment and personal information must be secure

**SR2.** Communication between the web app and Smart Serve must be secure

**SR3.** The operator must be responsible for correctly operating the machine and restocking ingredients

**SR4.** The user must provide identification and be of drinking age to order and receive a drink from Smart Serve

## **4.8 Cultural and Political Requirements**

Not applicable

## **4.9 Legal Requirements**

**LR1.** Smart Serve must only be used in countries where alcohol is legal

**LR2.** Smart Serve must not produce a drink that is consumed by persons under the drinking age

**LR3.** The operator of Smart Serve is liable for all drinks served

## 4.10 Health and Safety Requirements

**HSR1.** Smart Serve or the operator must not serve a drink to someone who is dangerously intoxicated

**HSR2.** Smart Serve should not be operational near unsupervised persons under the drinking age

**HSR3.** Smart Serve or the operator must not serve more than 2 drinks per hour to one user

**HSR4.** No electrical component can come into contact with any fluid

## 5 Likely Changes to Requirements

### 5.1 Machine Improvements

Although the first design of the machine has a degree of durability and ease of maintenance, a goal within the group is to have the machine not only be able to last longer, but allow for even less maintenance/care. This can be accomplished with the following new requirements.

**RLC 1:** Incorporate self-cleaning after every drink is made

**RLC 2:** Design system hardware to have a high water resistance

**RLC 3:** Have the system be more robust to lessen the risk of impact damage

**RLC 4:** Machine sends specific error codes to the operator to inform why the machine is breaking

### 5.2 Data Analytics

With the current design, the system will use basic Data Analytics to track information like popular drinks and stock of ingredients. A likely change will be to take advantage of the data to produce the following requirements.

**RLC 5:** Purchasing patterns are used to track optimal inventory

**RLC 6:** Analysis of user data is used to find correlations with drink orders to derive optimal drink mixes or drink recommendations

## 6 Unlikely Changes to Requirements

**UCR 1:** No new input of human interaction for base functionality.

**Justification:** One of the main purposes of this project is to incorporate more automation into the bartending profession. The requirements outlined reflect that. Any increase in human interference for base functionality will contradict this purpose.



**UCR 2:** Application will **always** comply with safety and the law.

**Justification:** Due to the device serving, it is important that all government regulations be followed. Misuse of the device could result in lawsuits, jail time, injury, or death. While handling alcohol, it is of the utmost importance that the device follows the substance's regulations.

**UCR 3:** Web Application must be user-friendly and very simple to use.

**Justification:** Theoretically, the system can be a substitute for the simplistic traditional human-to-human interaction of asking for a drink. The app must be at least as straightforward as the traditional method.

**UCR 4:** System must be able to create drinks with the high the level of preciseness and consistency as an average bartender.

**Justification:** With the system replacing the need for a bartender, it must be able to (at the very least) create drinks with both the appropriate measurements and a strong degree of consistency.

**UCR 5:** Maintenance of Smart Serve must be simplistic.

**Justification:** It must be assumed that a large demographic of operators of Smart Serve do not have a background in technology. The maintenance of the device should be straightforward, with limited technological experience required.

## 7 References

Not applicable.

## 8 Appendixes

### 8.1 Reflection

**Max Turek:**

As a mechatronics engineer, we get experienced with software, electrical and mechanical aspects but never become experts in one. Therefore, one skill I will need to learn how to acquire is working on mechanical and electrical systems specifically pumps for transporting fluids. Furthermore, I will need to understand front-end and web development more in order to launch and troubleshoot our web app that supports the user interface.

One way I find is best for acquiring new knowledge is youtube videos. It is possible to find a video explaining almost any skill now a days. The other channel of acquiring knowledge would to speak to alumni and friends that have completed a mechatronics capstone. I have many friends that are knowledgeable in software and hardware that are able to provide consultation and advice.

**Sam Nusselder:**

The team will need to have a strong understanding of mechanical, electrical and software components as well as the capabilities of integrating these systems. Mechanical and electrical systems related to delivering and controlling fluid flow will be required. Software knowledge with web-based applications will also be needed to develop an interface so users are able to interact with Smart Serve. Team management and group work skills will be needed in order to ensure a successful working team of 5 members. It will be important to make sure everyone always has a task to work on while also ensuring everyone understands the "bigger" picture and is aware of the long-term goals.

Some approaches to acquiring a strong understanding of mechanical, electrical and software components is to review any applicable course work and to do more research online to learn about these topics. All team members will need knowledge in all of these areas, so they will need to pursue which ever approach works best for them as they feel they need a stronger understanding. To acquire better team work skills, team members can look past on their individual previous group work experiences and learn from things that went well and things that didn't. Another approach to improve team work skills is to speak with other current groups to learn and understand how they work effectively as a team. All team members can pursue the approach that works best for them. Not everyone has great team work experience to learn from, so this is where the second approach would be more effective.

**Peter Minbashian:**

As a computer engineering student at McMaster, a majority of my foundational knowledge, surrounding technology, revolves around embedded software. Although that is applicable to this project, I present not much value to the web development portion. Therefore, it would be best that for me to learn more about web development. Not only will this entail learning the basics (JavaScript, HTML, CSS) but also learning specific tools required for this project. This includes how to incorporate user log-in functionality on a website, connecting a website to a server, and connecting a website to an API. I believe I can learn this skill through two different avenues: Udemy and my friends/family.

Although there are a plethora of free resources on the internet, I often learn new skills using Udemy. The website has a course for everything, at very affordable prices. The benefit of using the site is displayed in the quality of the instructors and mini-projects found throughout the course to apply the knowledge learned.

Another resource I intend to use is the help of others. I have many close friends who work within the web development space along with my parents. I will not shy away from asking questions to gather a deeper understanding of the field.

The combination of both Udemy courses and my personal resources will allow me to gather a

deeper understanding of web development and be a stronger asset to my team.

**Ryan Were:**

Important knowledge needed to successfully complete our capstone project will be the integration of software and hardware. Understanding how we will be communicating between the two will be crucial in getting the brains of our project to do the physical work. An important skill to have will be organizing specific knowledge champions in the group. This means that one person will be a subject matter expert on a certain topic and be able to mentor the rest of the group members in that area if needed.

To master software and hardware communication, one approach would be to review previous information provided to me in courses. Brushing up on what I have previously learnt will help reinforce this knowledge. Doing my own research online about this topic will also help greatly. It will allow me to develop a further understanding on specific areas that I may struggle in. Another approach that I could take is emailing a professor to see if they are willing to answer questions that I may have about this topic. Having an expert on the topic answer specific question I have would give me the exact knowledge I need quickly. The approach I'd take first is to do my own research. I think that in a workplace in the future, being able to find info you need on your own is a valuable asset to have. Regarding organizing knowledge champions, one approach to take is to have a group meeting where everyone discusses what they consider themselves knowledgeable in, and deciding what topics we are lacking in. That way we can organize the group efficiently to cover all the areas that we must be masters in to complete the capstone project. Another approach would be to just blindly assign people to certain topics. The first approach of communicating with the team sounds much more efficient and reduces redundancies, so I would take this approach. I believe this is important because communicating with people about their strengths and weaknesses and developing a strategy to get the most out of the group as a whole is a good skill to have for anyone who wishes to go into management.

**David Bednar:**

As a software engineering student working on a project with a large mechanical and electrical component, I will need to pick up some skills from both fields. I will be focusing mainly on the software side of the project, but I will learn mechanical and electronic skills relevant to the project before beginning so that I can help work on this. I also have most of my experience in backend software engineering and working on frontend is something that I have done but do not have the most experience in.

I plan on working together with my mechatronic group members to learn how to build different mechanical components, and connect electrical wiring. I will also get help from a colleague of mine who is currently working in a mechanical engineering role to learn how to use certain tools and machines. To learn frontend tools like HTML, CSS, and javascript, I will use online tools such as youtube and Udemy. I will also work together with my group mate Peter to learn about frontend principles and build up the frontend of the project together.