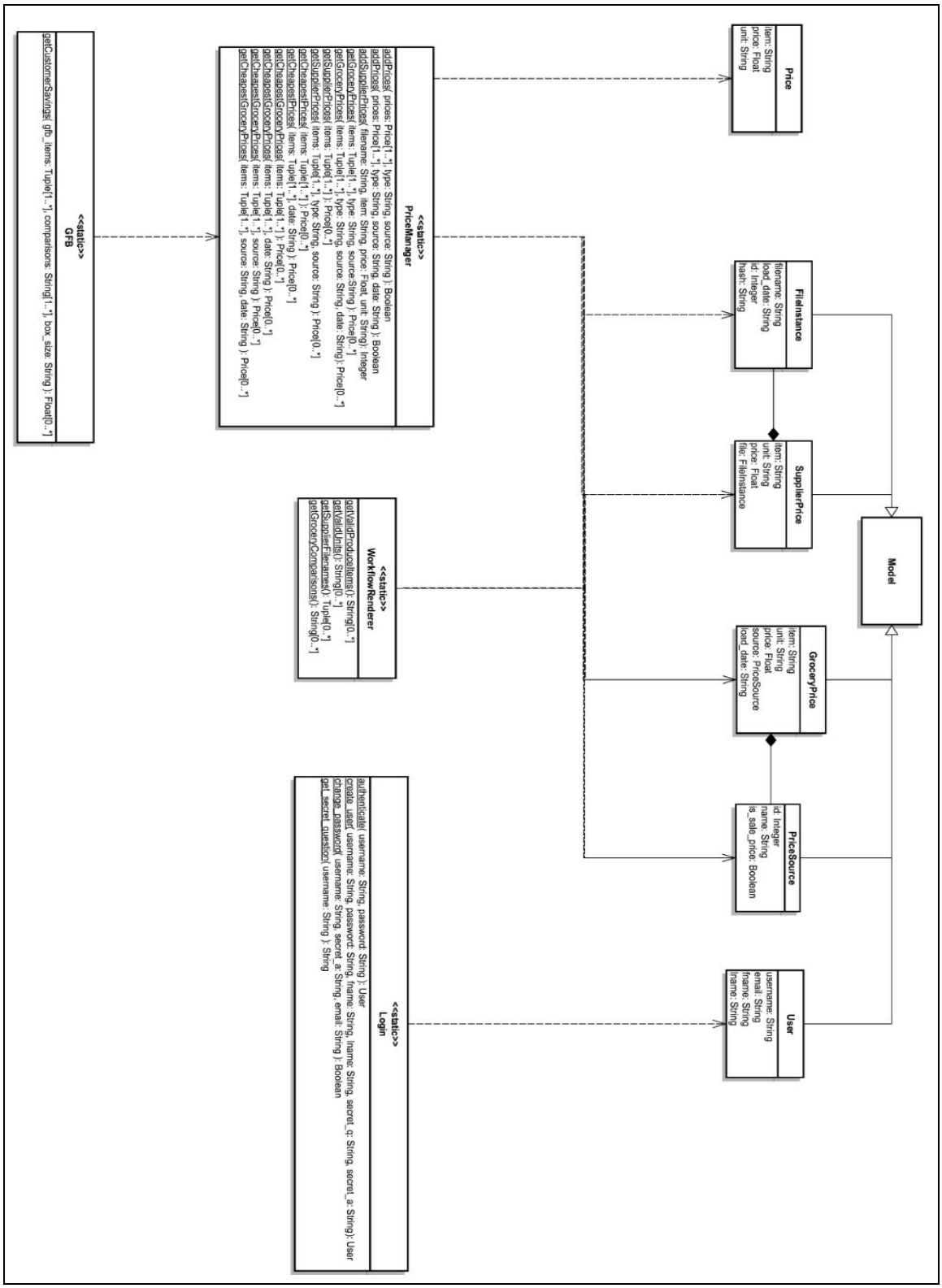
A. Class Diagram



Justification

In our class diagram, we have used composition over aggregation as a relationship between some of the classes. This is illustrated in the diagram between the “FileInstance” and “SupplierPrice” classes. Moreover, the composition relationship also holds for “GroceryPrice and “PriceSource” classes.

As a super group we decided that the “FileInstance” class does not exist without “SupplierPrice”. This is because the user provides the supplier price list using a file dialog box in the upload page. The file dialog box creates the instance of the file based on the provided supplier price list. Hence, without a supplier price list, the file instance cannot be created as it depends on it. Thus, this is indicated through the use of the composition relationship.

The similar reasoning applies to the composition relationship between the “PriceSource” and “GroceryPrice” classes. In this case, the price source is dependant on the grocery price list. Without a list of groceries, a price cannot be computed to be used in calculating savings for the Garden Fresh Box and POD flows.

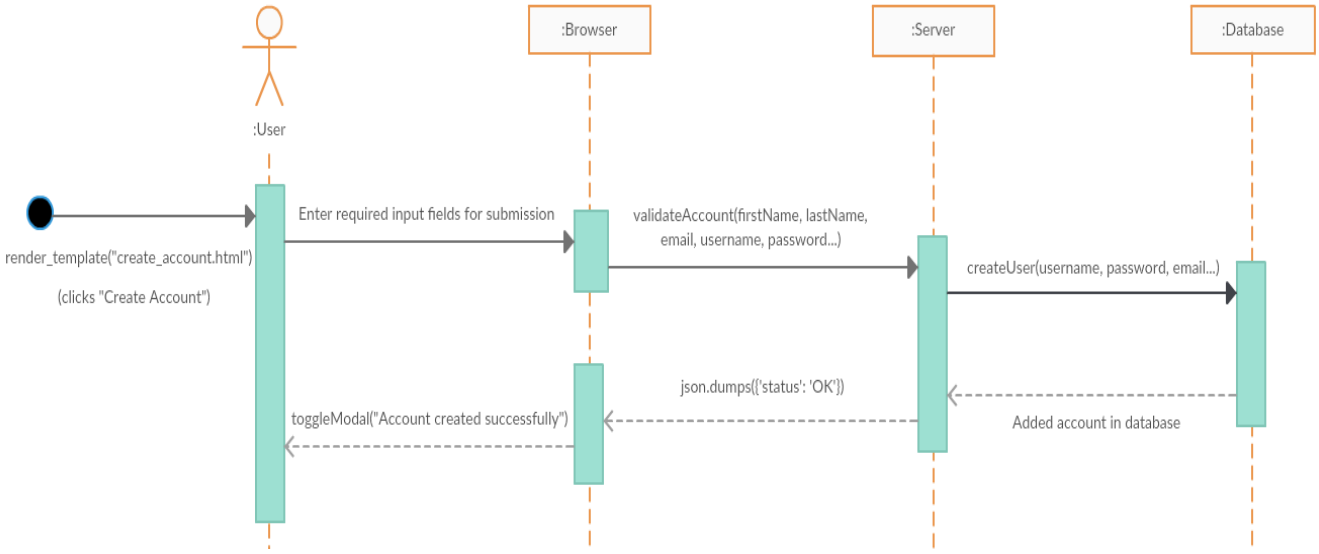
Link to Full Sized Class Diagram

https://drive.google.com/a/uoguelph.ca/file/d/0B-GVZTaYJ7LSX3dUSzhMZjhaTVk/view?usp=sharing

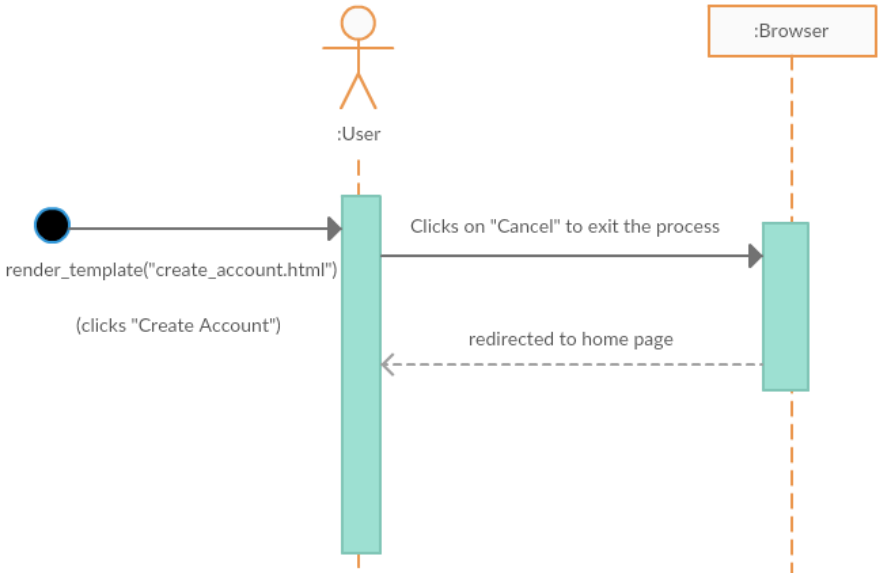
B. Sequence Diagrams

Use Case 1 - Creating an account

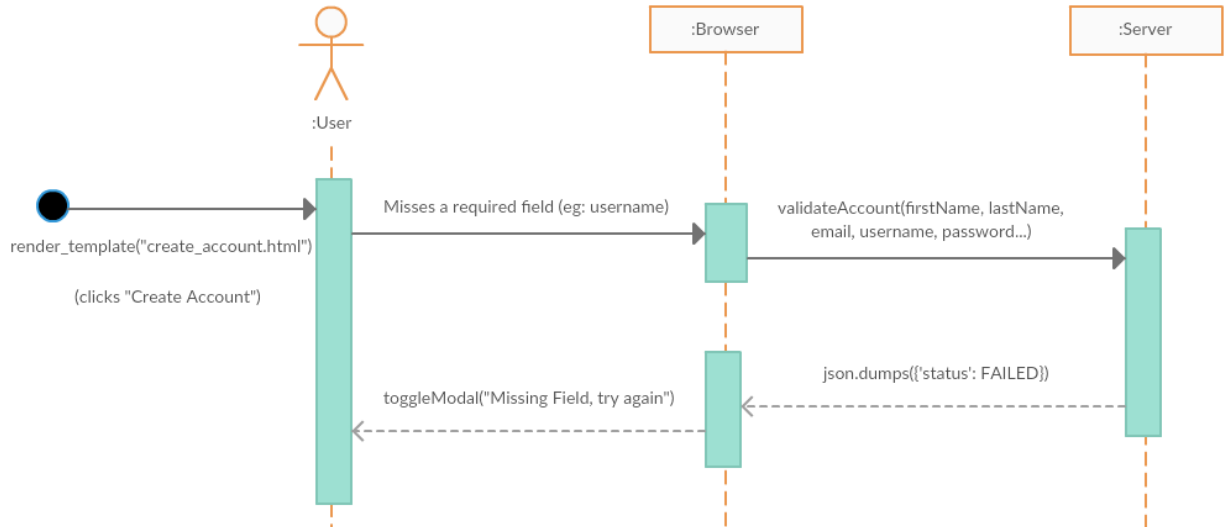
Main Scenario:



Alt 1 - The user cancels the request to create an account:



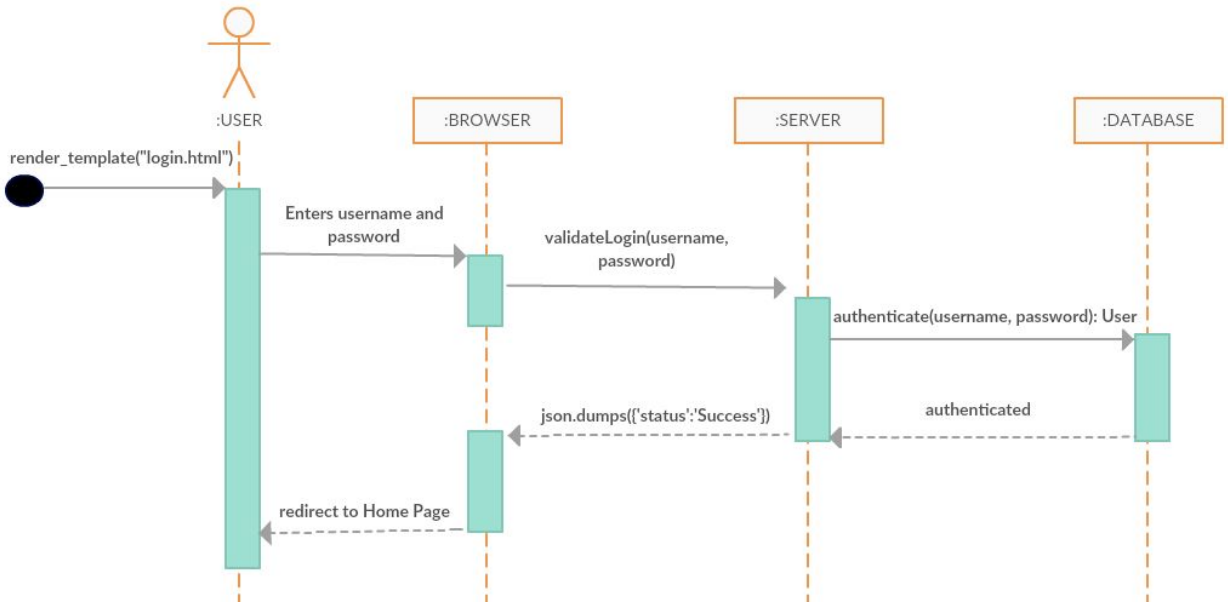
Alt 2 - Data entered was incorrect, missed required field:



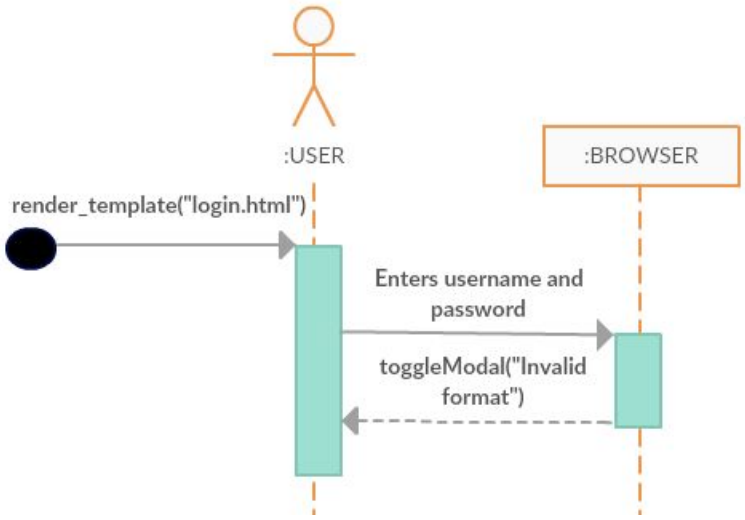
Note: For the case of Alt 3, the functionality to send the verification link to the user for the create account process was removed in the final prototype. Hence, there is no implementation in place for that step.

Use Case 2 - Login to the system

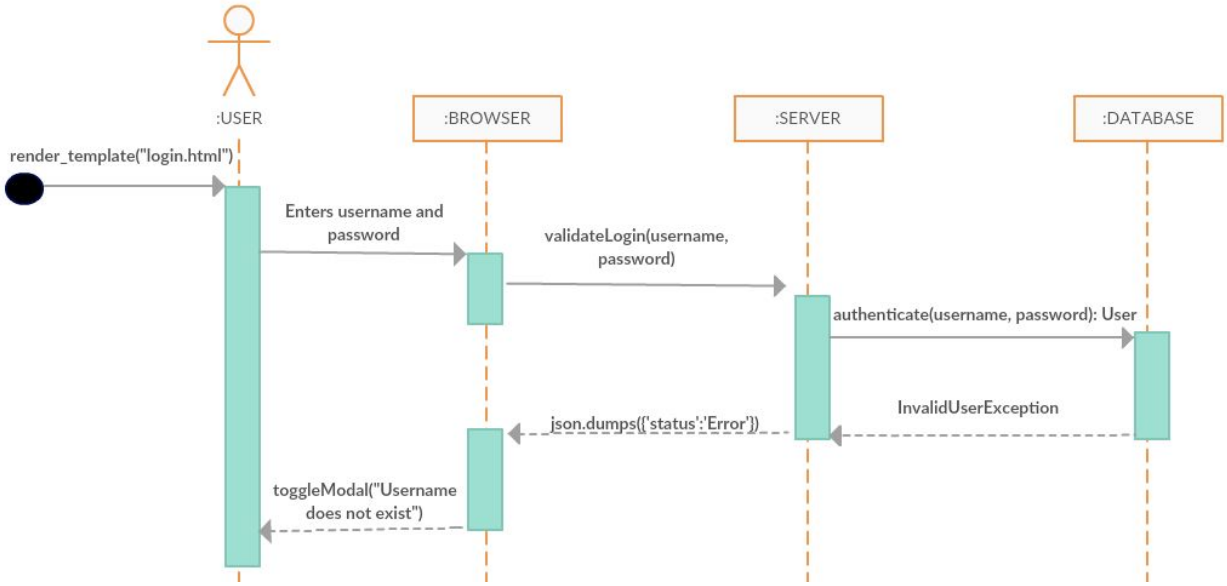
Main Scenario:



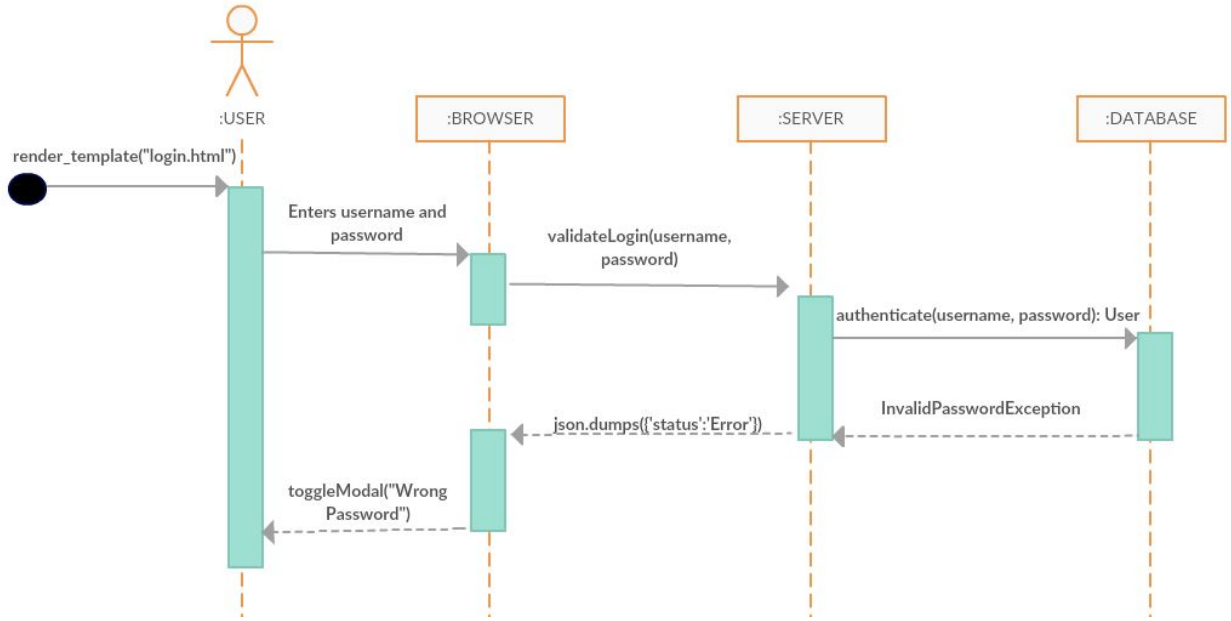
Alt 1 - Invalid input format:



Alt 2 - Username does not exist:

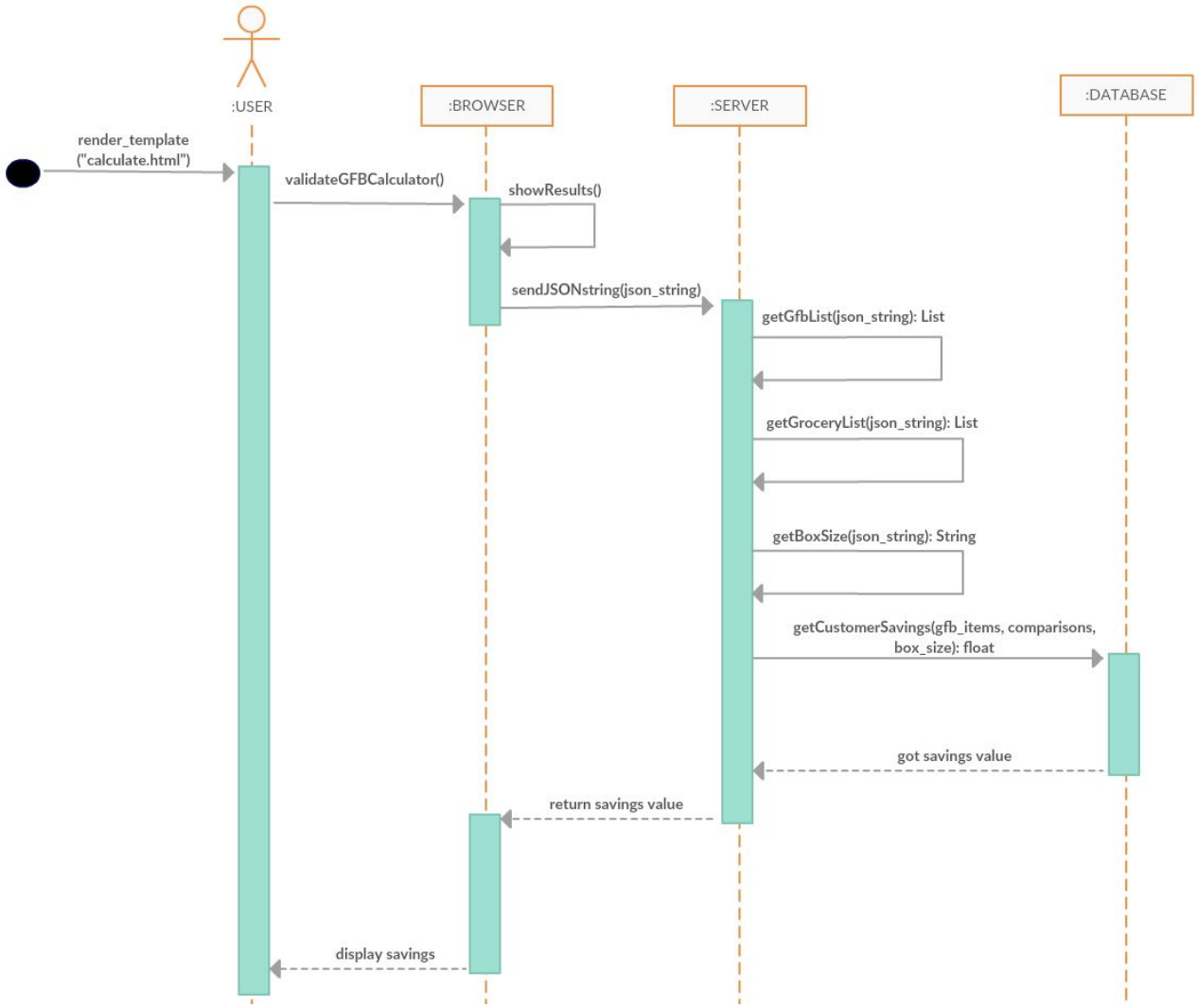


Alt 3 - Invalid password:

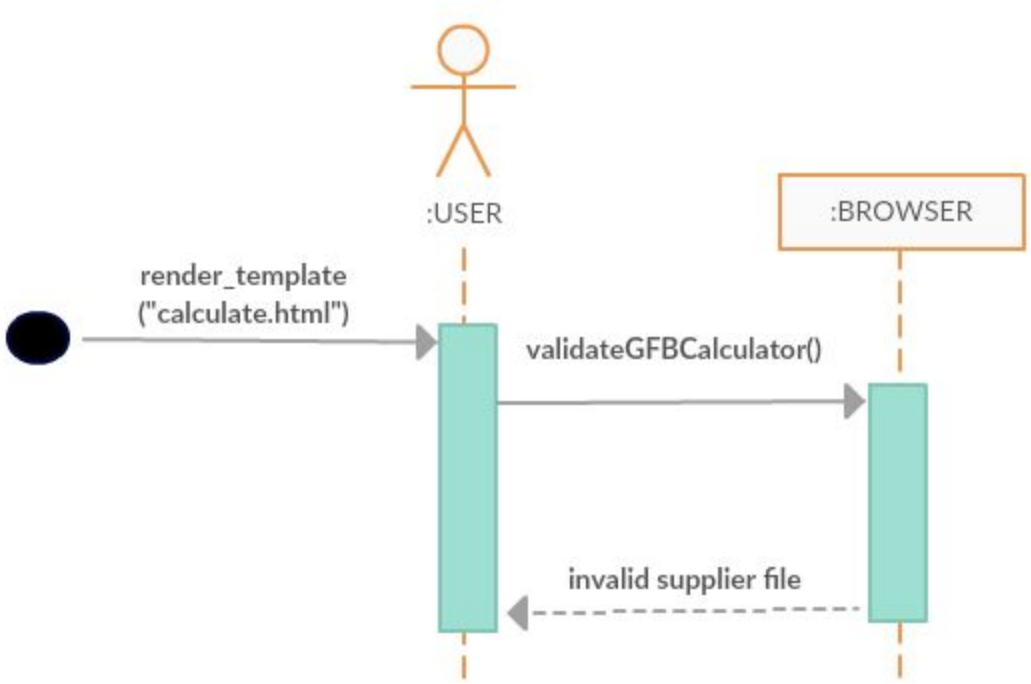


Use Case 3 - Calculate savings

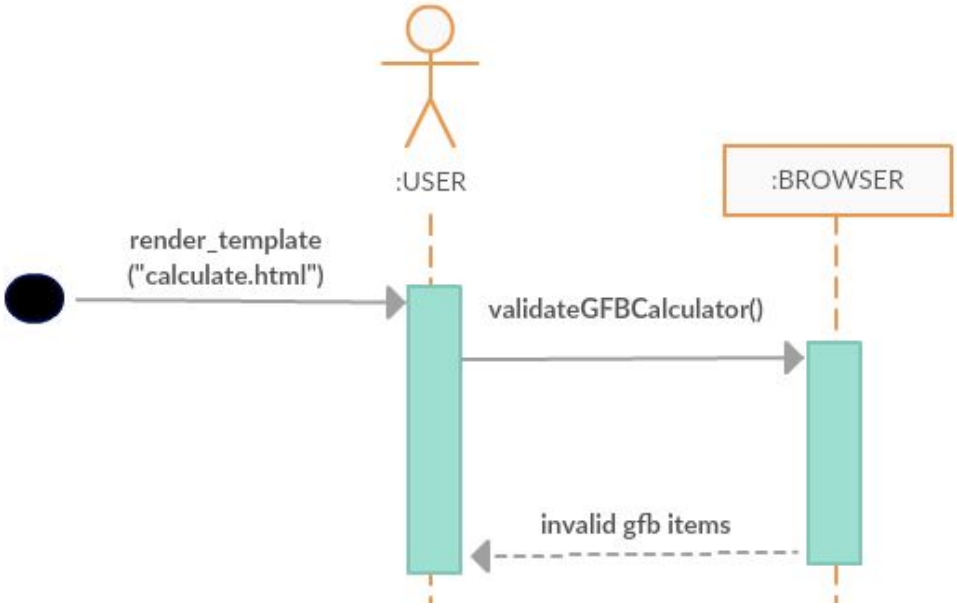
Main Scenario:



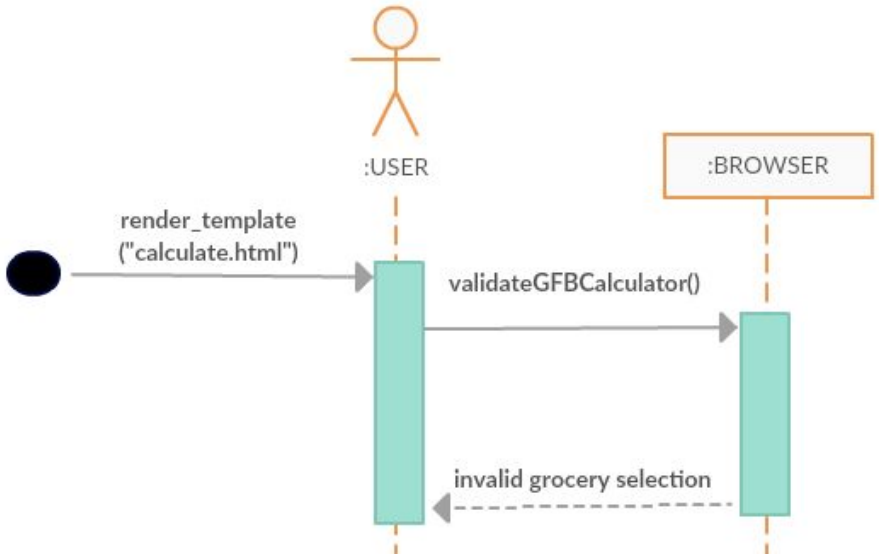
Alt 1 - Invalid upload file:



Alt 2 - Missing Garden Fresh Box items:



Alt 3 - Missing grocery stores for comparison:



C. Next Steps

- **Remaining Requirements:**

User	Category	Requirement	Dependency	Priority	Time Estimate
System	Must	The system must have a data store abstraction layer that can convert all date values to project standard date values before storing them in the data store.	1	10	0.5
System	Must	The system must calculate the total retail price of the GFB if purchased from each competitor.	32	10	0.5
System	Must	The system must calculate the difference between the actual cost of the GFB and the GFB if purchased from each competitor.	35	10	0.5
User	Must	The user must be able to download the completed third party price list template to their local machine.	47	10	0.5
System	Must	The system must delete all information from the data store 20 years after the date of entry.	1	20	0.5
User	Must	The user must be able to enter a custom price to remedy situations when the requested price for a specific produce item from a	36	20	1.5

		specific supplier has no matching entry in the data store.			
System	Must	The system must be able to the show the user what price sources from what date are involved in a savings calculation.	59	20	0.5
System	Should	The system should determine the percent savings generated for a customer when they purchase a GFB versus purchasing the equivalent produce at local low-tier grocery stores.	22	30	1
System	Should	The system should determine the percent savings generated for a customer when they purchase a GFB versus purchasing the equivalent produce at local mid-tier grocery stores.	22	30	1
System	Should	The system should determine the percent savings generated for a customer when they purchase a GFB versus purchasing the equivalent produce at local high-tier grocery stores.	22	30	1

System	Should	The system should scrape the competitors address from their website.	2	40	0.5
System	Should	The system should scrape the competitors phone number from their website	2	40	0.5
System	Should	The system should alert the user if the connection to the data store web service is lost.	70	40	0.5
System	Should	The system should alert the user when the data store has been restored from a backup.	72	40	0.5
System	Could	The system could allow the user to select which competitors prices to compare the GFB cost to.	3	50	1
System	Coould	The system could allow the user to select which competitors prices to compare to the POD's prices.	3	50	
System	Could	The system could attempt to predict when produce items are going to come on sale.	20	50	2
System	Could	The system could calculate price trends for a specific produce item from a specific competitor.	22	50	0.5

User	Could	The user could select which stores they want to be included in the competitor pricing average so that they can control which stores are compared in the savings data.	24	50	0.5
System	Could	The system could calculate price trends for a specific produce item from a specific supplier.	24	50	0.5
System	Could	The system could compute the optimal GFB based on supplier pricing.	24	50	1
System	Could	The system could display the contents of historical GFB's.	67	50	0.5
System	Could	The system could have an allow users to reset their password via e-mail if they forget it.	82,88	50	2
System	Could	The system could provide graphs to show the user price trends for a specific produce item from a specific competitor.	92	50	1
System	Could	The system could provide graphs to show the user price trends for a specific produce item from a specific supplier.	94	50	1

Time Estimates:

Table 1: Timeline for Remaining Requirements

halfday	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
day	**1**	**1**	**2**	**2**	**3**	**3**	**4**	**4**	**5**	**5**	**6**	**6**
Iteration	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
P1	8	56	56	56	91	91	91	91				
P2	37	60	84	85	92	94			97	97	97	97
P3	41	74	74	86	93	95	95					
P4	48	75	75	87	90	90	100	100				
P5	50	76	76	89	89	96						

Assuming that there is a team of 5 working full time hours with a velocity of 80% to complete the remaining 24 requirements of the Garden Fresh Box savings calculator it would take approximately 6 days to complete the project in its entirety. Given that each of the 5 developers would be making $1200 a week to work on the project the total cost of completing the system can was calculated to be $5850 as follows:

$$Cost\ Per\ Developer = \frac{Days\ worked}{Days\ in\ a\ week * Velocity} * 1200$$

$$P1\ cost = \frac{4}{5 * 0.8} * 1200 = \$1200 \qquad P2\ cost = \frac{5}{5 * 0.8} * 1200 = \$1500$$

$$P3\ cost = \frac{3.5}{5 * 0.8} * 1200 = \$1050 \qquad P4\ cost = \frac{4}{5 * 0.8} * 1200 = \$1200$$

$$P5\ cost = \frac{3}{5 * 0.8} * 1200 = \$900$$

$$\begin{aligned} \text{Total cost} &= P1 + P2 + P3 + P4 + P5 \\ &= 1200 + 1500 + 1050 + 1200 + 900 \\ &= \$5850 \end{aligned}$$

F. Individual Contributions

- **Alexandra Campbell:**

One thing I really enjoyed about CIS*3750 was how the projects actually have a real world purpose. Too often in cis classes we are asked to create different programs that are implemented for the sole purpose of us gaining understanding about a certain concept that we have been learning in class, but in this course the project that we were expected to implement was something that would help the Guelph community attain food of nutritional value so that they could live a happier and healthier life. This aspect of the course had me completing each project for a purpose other than attaining a certain grade, which was a nice change.

One of the things I disliked the most about this course was how the lecture times were more about group work and testing our knowledge than actually providing us with the necessary knowledge to do well in this course. As I personally learn better when I am being taught rather than when I read something myself, I believe that having a traditional lecture would have allowed for me to come out of the course having learned more than I feel I have thus far.

Something that I would like to see change about this course would be that the students are expected to teach themselves a majority of the content for this course in their own time. Although the university sets a certain amount of work time outside of class for each course not everyone has that much time to read articles and listen to podcasts every week and this can have a negative effect on quiz and exam results. I think that in the future more of the course content should be taught within the lectures so that we all have the basic information to succeed in this course.

- **Dominic Lee:**

I thought this course made a good attempt at providing us with the experience involved with the various stages in software development. The thing I liked most about this course was the opportunity to work with real life community projects. This offered a refreshing change to my previous courses, as we not only were able to learn a multitude of development concepts, but were able to apply them to a project that could possibly have a huge impact on our immediate environment. I thought the bi weekly goals, such as requirements gathering, prototyping sessions etc., reinforced the idea the we were working on something real, which could eventually lead to a longer termed project attracting even more likeminded individuals.

One thing I didn't like that much were the infrequent meetings with the client/s. I think if we met with the client more often, many of the uncertainties that were encountered could have been cleared up much more quickly. This would of allowed us to have more time to improve the final product, which would have resulted in a better application.

If I could change one thing in this course, I would modify the desired learning outcomes offered in the lab sessions. The lab sessions were more or less free sessions that expected us to use them to prepare for each bi weekly goal. I think they could've been more productive, and perhaps allowed us to work on the coding portion of the application, or even on unrelated coding tasks using the prevailing technologies of today.

- **Ethan Nichol:**

The thing I loved about the course was the trust and autonomy places on us as individuals. We still had to follow the course curriculum in terms of our design process (i.e. paper prototype->wireframe->prototype) but we were trusted with the choice of how we organized our group and supergroup within ourselves. This was especially fulfilling because we were working on a real product that someone is going to use, and not just a throwaway assignment, so the choices made in the design have actual implications.

There isn't really anything that I hated about this course; The marking and workload was fair, the content was both relevant and engaging, and the group work went far smoother than in some other courses. All in all this was probably the best course I took this semester. One minor gripe I do have is the stupid wheely chairs in the classroom. They don't really help you move or work in groups when they are all in constant gridlock.

The thing I would change about the course is to give better structure to the supergroups. Our groups organized well together and ended up with a solid product, and all the groups were pretty good about staying in touch in Slack, but I can imagine a scenario where a group within the supergroup could easily go dark and there's nothing the other 2 groups can do about it. It seems a bit risky to have upwards of 15 people organized on one project when communication can be very scarce in some cases. I realize that regular check ins with the groups and documentation such as shared requirements and API docs help offset this, but I feel that forcing the groups to do a regular check in (in the lab times maybe) in the form of a supergroup standup could be beneficial, since not everybody is going to be able to meet during class time.

- **Joshua Pinsent:**

One of the things that I loved about this course was being able to interact directly with the client and receive direct feedback on our prototype interface designs. I have never been able to experience anything remotely similar to this aspect of the course in any of my other classes, and having had the opportunity to work with clients over the course of my Co-op work terms, I can confirm that the client interaction that this course offers very closely resembles the experience of working with clients in the workforce. The presentational and prototyping skills that I learned throughout this class will certainly be helpful in my future endeavours working as a developer.

One aspect of the course that I did not enjoy was the reverse-classroom style of the lectures. I feel as though I would have learned many of the course concepts better if the lectures were more traditional. The amount of reading we had to do to prepare for the group quizzes in the lectures was something I had a difficult time keeping up with, being unable to do a majority of the readings until the weekend. This left me unprepared for the group quizzes, leaving me unable to assist my group and unready to learn more about the concepts that I hadn't yet been able to familiarize myself with.

If I could change one aspect of the course, I would change the reverse-classroom style of the course and make the lectures more traditional. I learn faster and more consistently from listening to a professor talk about concepts during scheduled class time, and I felt as though the group quizzes were a waste of time I could have spent taking notes and learning the course concepts.

- **Ranbir Sihota:**

I loved that lecture time was a more relaxed and more engaging compared to other classes because I was spending less time frantically taking notes and more time taking in the content being presented. For other classes, most of the time the professor would just lecture for the whole time and I would have to take notes so that I could revise the material later. With this class, the professor presented us with questions related to the course and as a group we would answer them. And the pace at which the answers were presented to us made it easier to take simple notes to prepare for the final exam.

I hated that there was not enough time to finish the product because while presenting the final prototype, it just felt like if there was more time to work on the product, there would have been more functionality to present to the client. For example, our super group decided to at least finish the login process and the Garden Fresh Box functionality. And if there was enough time, we would finish the POD functionality. It turned out that we only had time to completely finish the login process and partially finish the Garden Fresh Box functionality.

I would change the fact that we must listen to podcasts and read the textbook outside of class time because along with assignments, developing prototypes and preparing for presentation of the prototype, this course really challenged my time management skills. I feel like the podcasts and lecture were sufficient in covering course material. Specifically, the podcasts were short enough to not completely make me lose interest in the topic being presented. And it is much easier to listen to podcasts and follow along with the slides than reading a textbook.

- **Shahrukh Suhail:**

During the semester, this course had exposed us to different tools and techniques that are extremely useful in a real life scenario upon graduation. One such example is the improv training at the beginning of the semester which focused on improving interpersonal skills, team collaboration and client interaction. As a co-op student, I believe these are important skills to learn and develop when working in large teams and dealing with real clients. Personally, I loved the inclusion of the improv training for this course. From a perspective of a co-op student, it teaches you how to interact with a client effectively in an engaging manner. Moreover, the training was also useful to introduce yourself to others and enhances networking skills. These skills are not only useful in class, but can be used on part time jobs, co-op work terms etc where client interaction and working in teams with effective collaboration plays a vital role.

On the contrary, I did not like the amount of time that was given for the final implementation of the product. In the course, we were only given two weeks to completely focus on our final prototype. Since our team was implementing the user interface, we had to wait for the other groups to finish their backend and scrapping components which would then be stitched together with the user interface. This led to implementing the code until the last minute and did not give a sense of pride upon completion due to the lack of some functionality and testing. Lastly, I would like to propose a networking event held with the client for future classes. This would allow each supergroup to have consistent requirements and get to know their clients further. Moreover, this would be an excellent opportunity to apply the skills learnt from the improv training and benefit from them further.