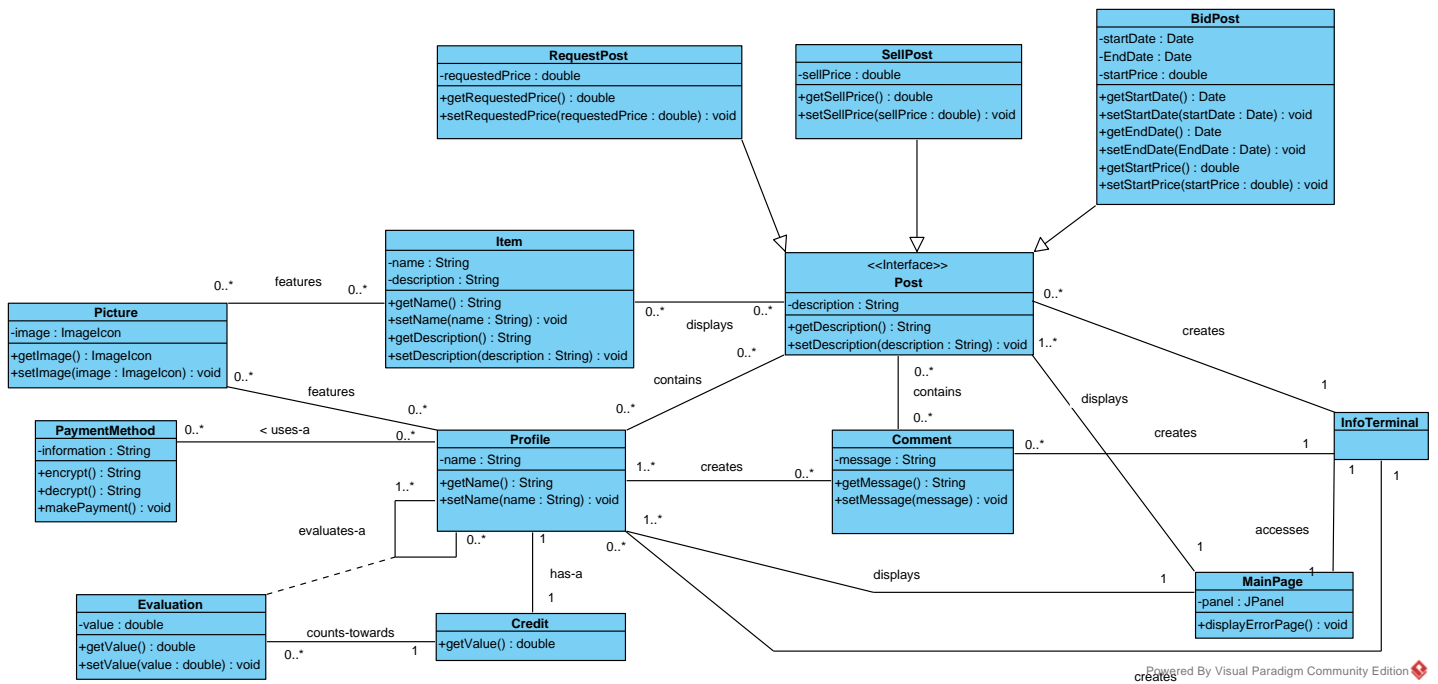


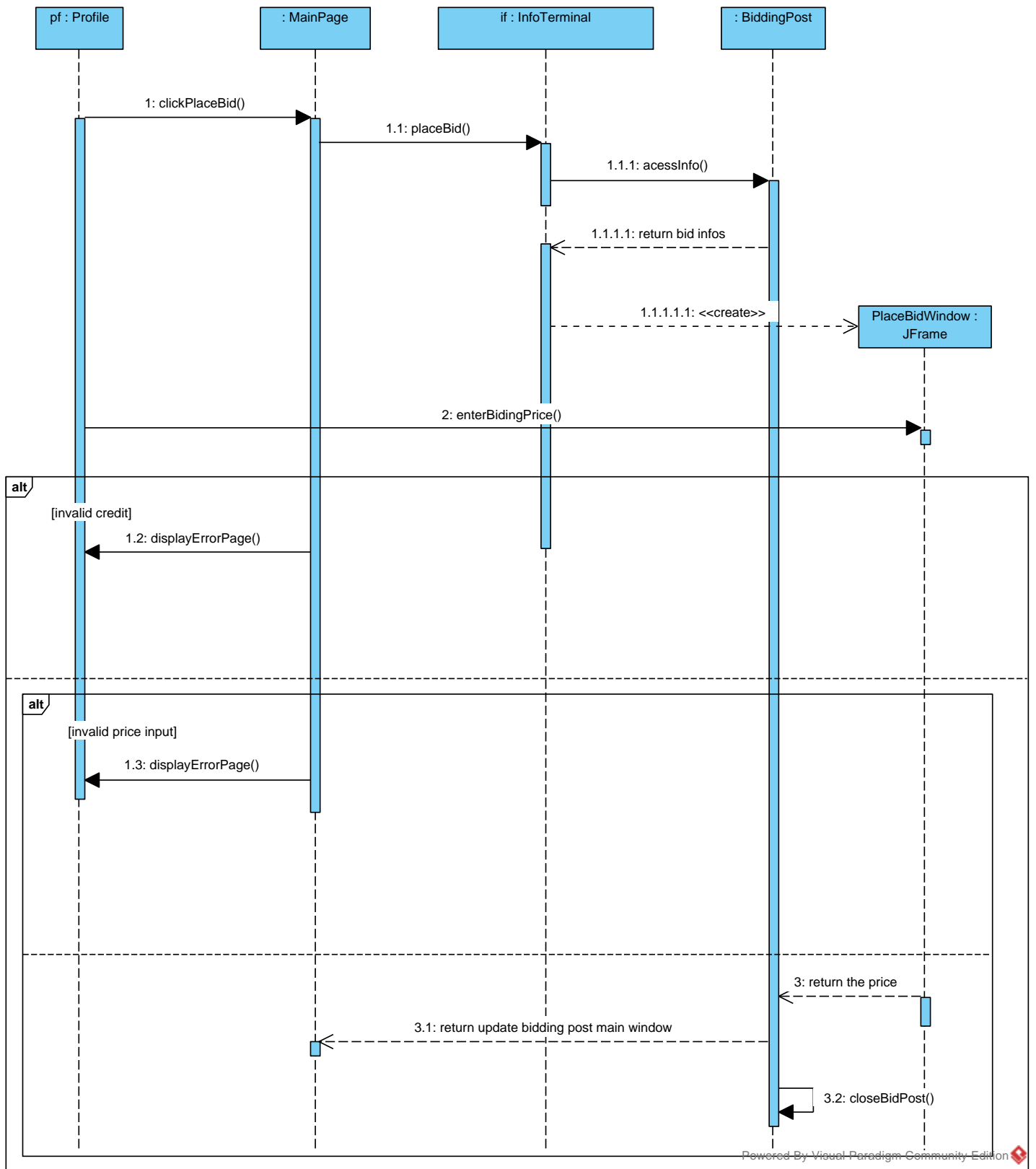
GRASP Analysis

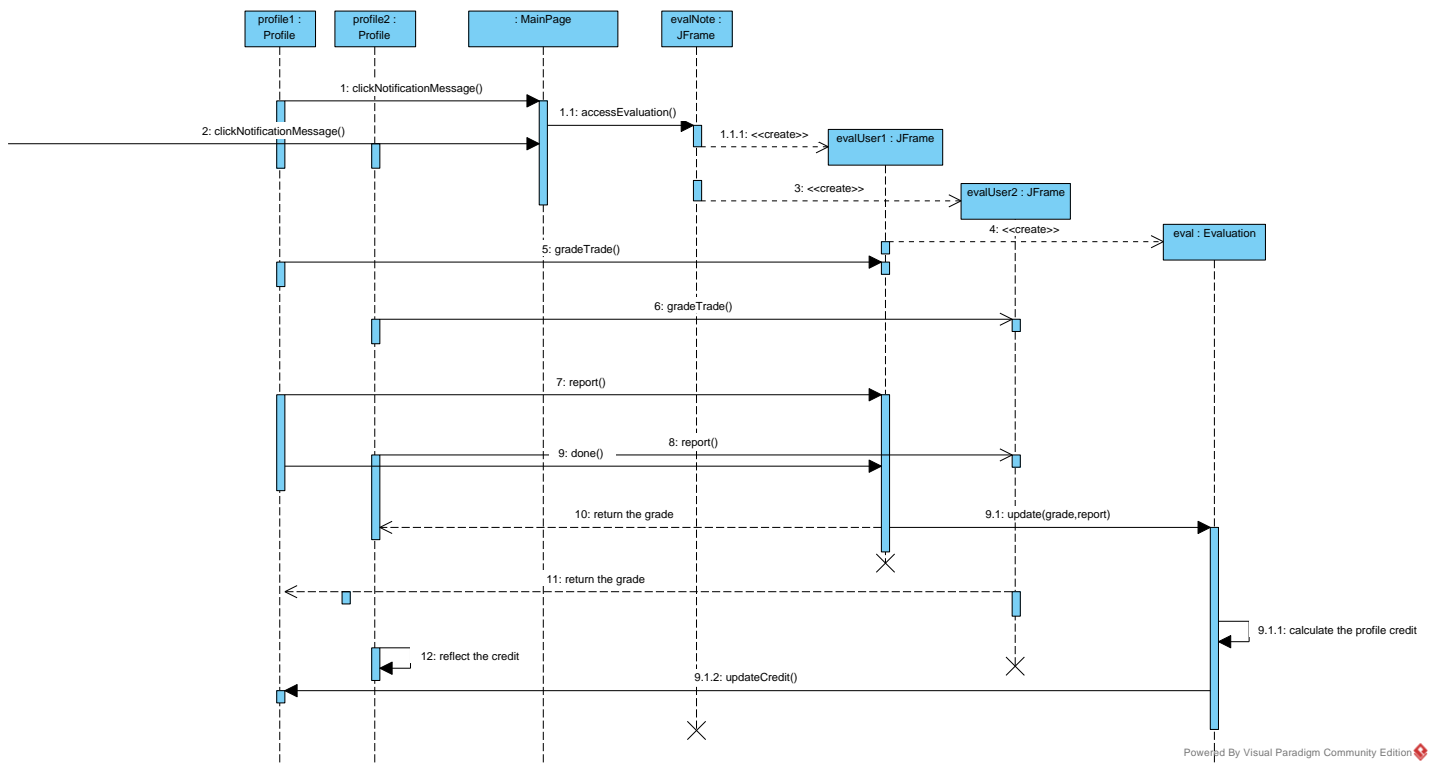
- If an object has a simple data type or predefined Java object stored inside, it is the information expert for those simple data types or Java objects and must be responsible for them. Therefore, getters and setters or other modifying methods for these simple data types exist within the classes that store them.
- The InfoTerminal object serves as the Controller for the entire application, acting as the system.
- The InfoTerminal is also the Creator for complex objects such as Profiles, Posts and Comments.
- The Profile serves as an Information Expert for its Posts and Payment Methods, which is why they will have lists to store their Posts and Payment Methods.
- Profiles will serve as the Information Expert for their associated comments since it would make sense for a post to keep track of all its listed comments. For this reason, Profiles will contain a list of Comments. Although the Post will also contain a list of comments that it references, this is for ease of access to its comments, not because a Post is an Information Expert on its comments.
- We have three separate polymorphic types of Post objects, to reduce coupling on the regular Post object. This was done because all Posts discuss an item in some way and need a description of said item. However, what each type of post does about said item is unique, so separate Post types are needed.
- The Credit and Evaluation objects are separated to lower the coupling for the Credit object. Credit is only responsible for displaying itself to its associated user, whereas Evaluations are responsible for modifying the total credit.

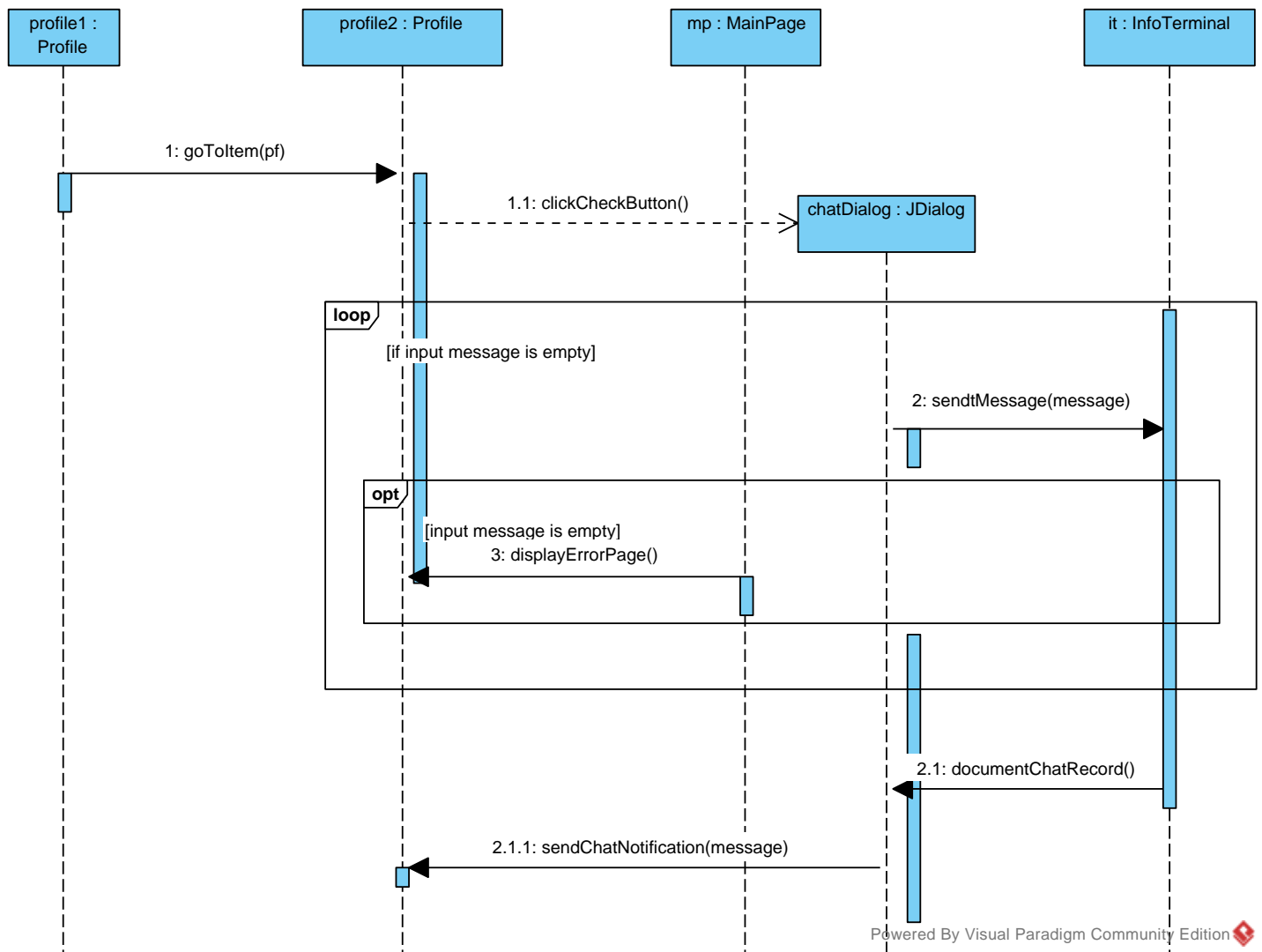
Test Coverage Plan

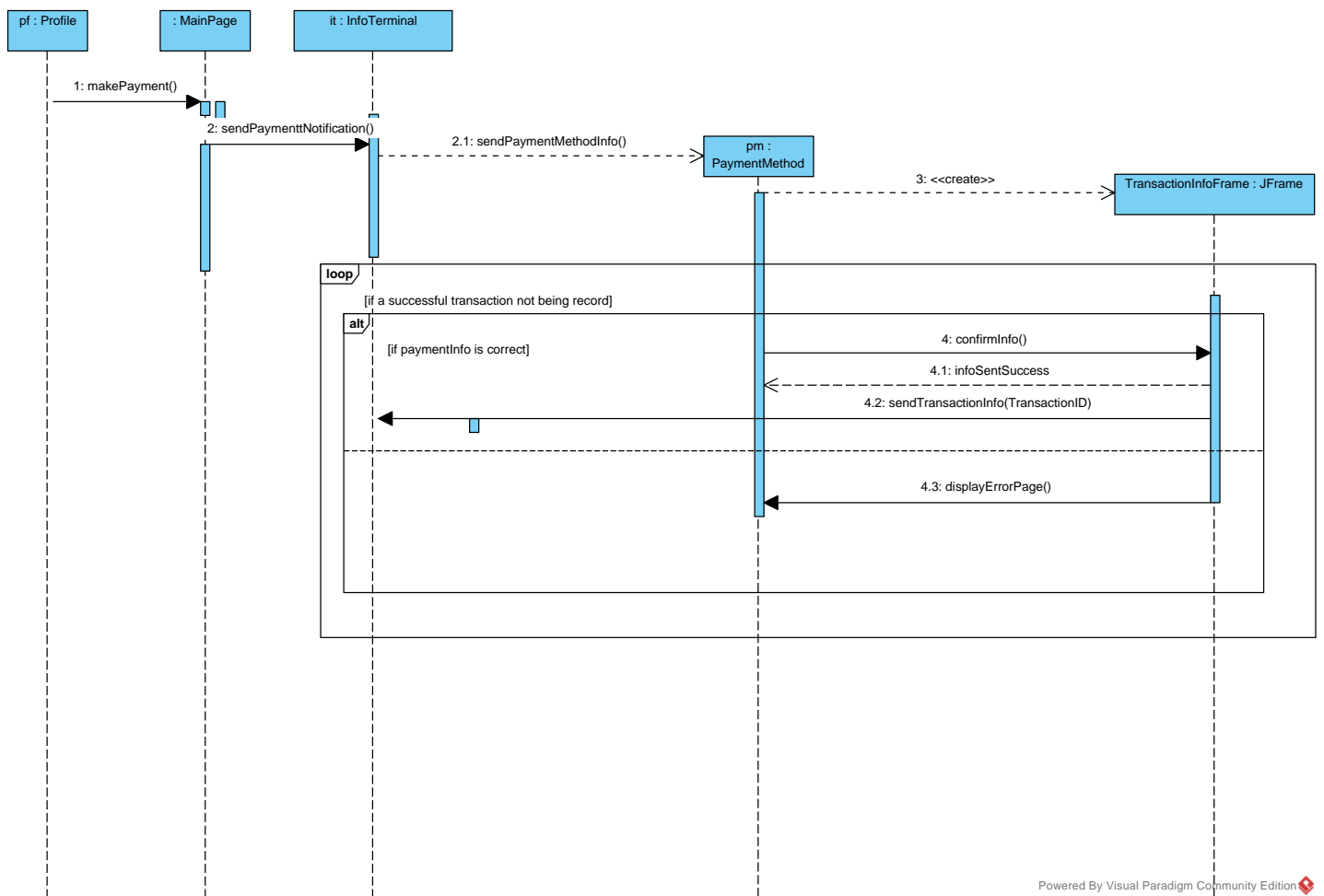
Our test coverage plan covers two approaches, one is using the JUnit test, and the other one is using WindowBuilder. After all the classes extrapolated from the class diagram are created, we will write the JUnit test for the methods in each class. We will run the JUnit test to make sure the methods are performing what they should do. The WindowBuilder on Eclipse will be checking how the GUI looks like for each window, frames, and dialogues we created to maximize the quality of the layout.

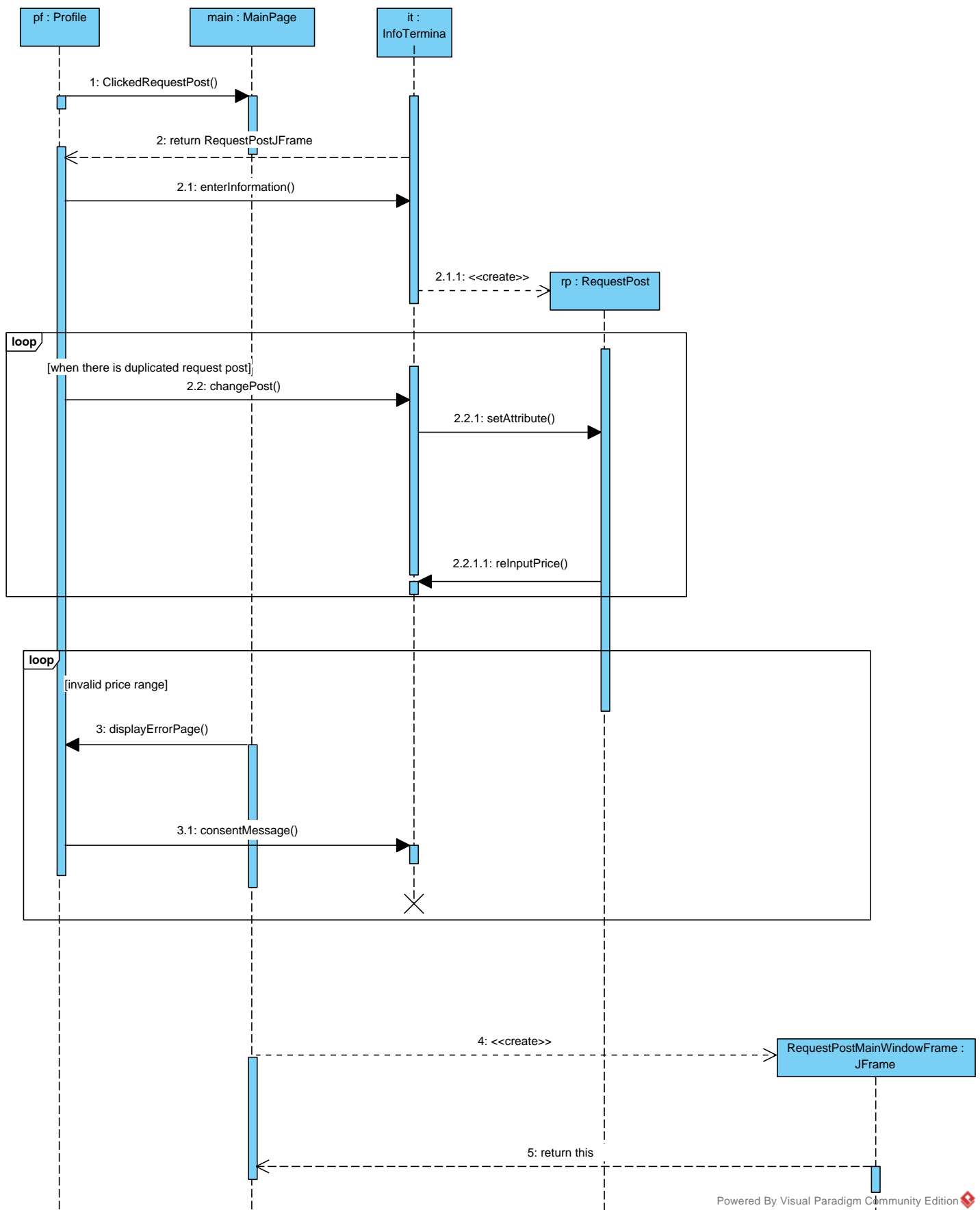


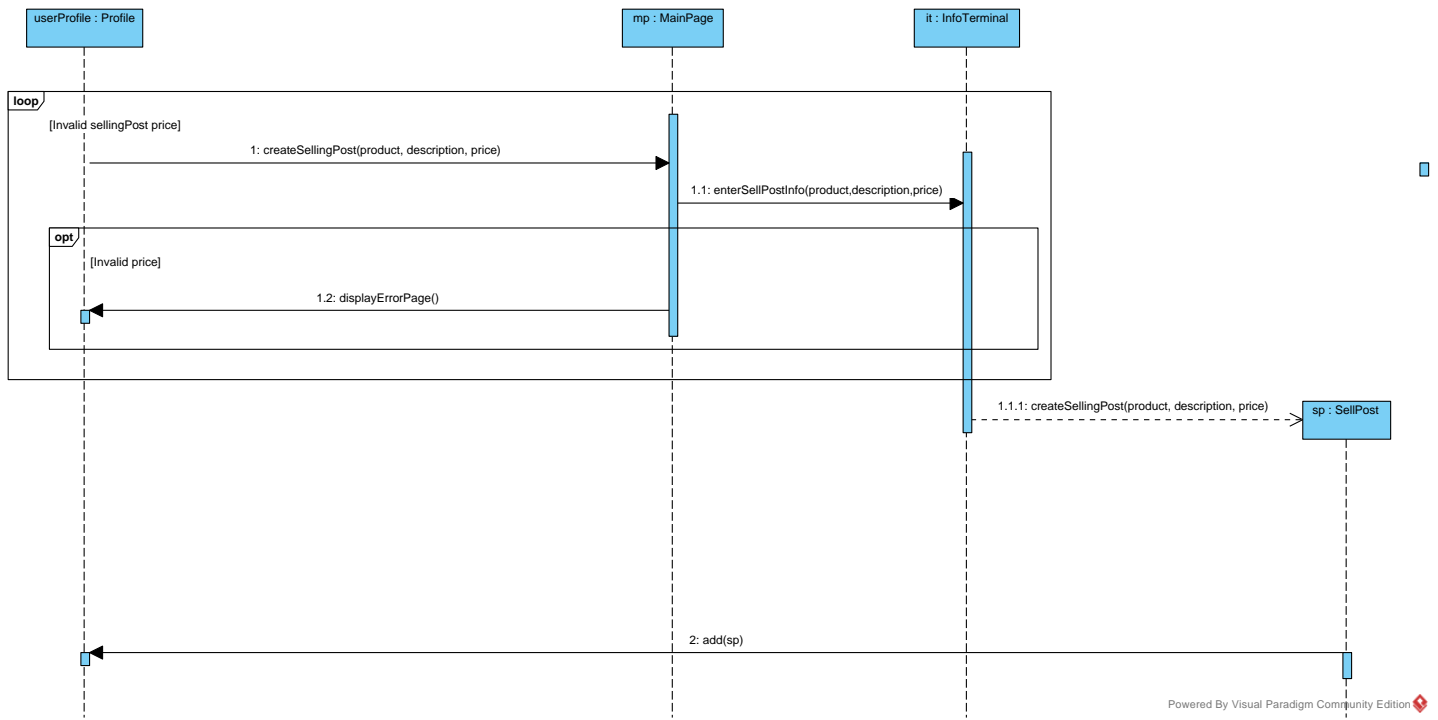


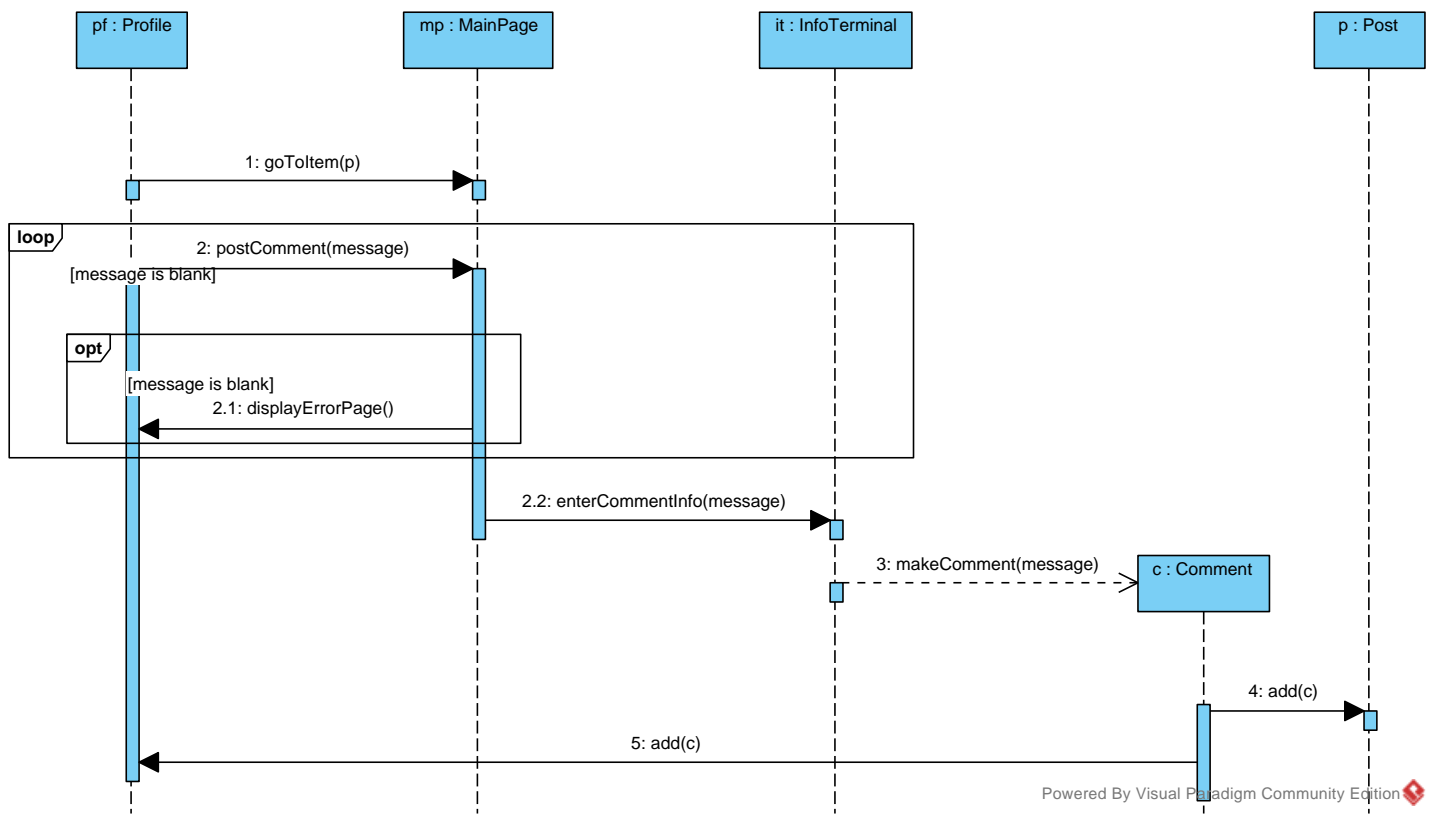


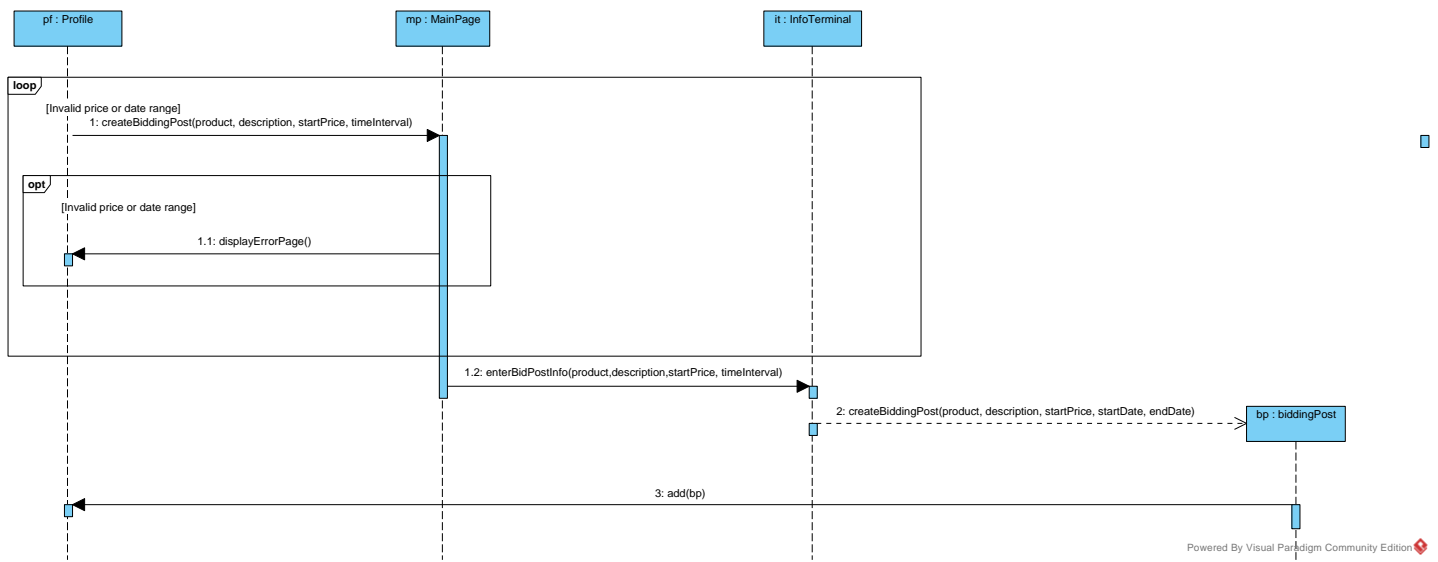


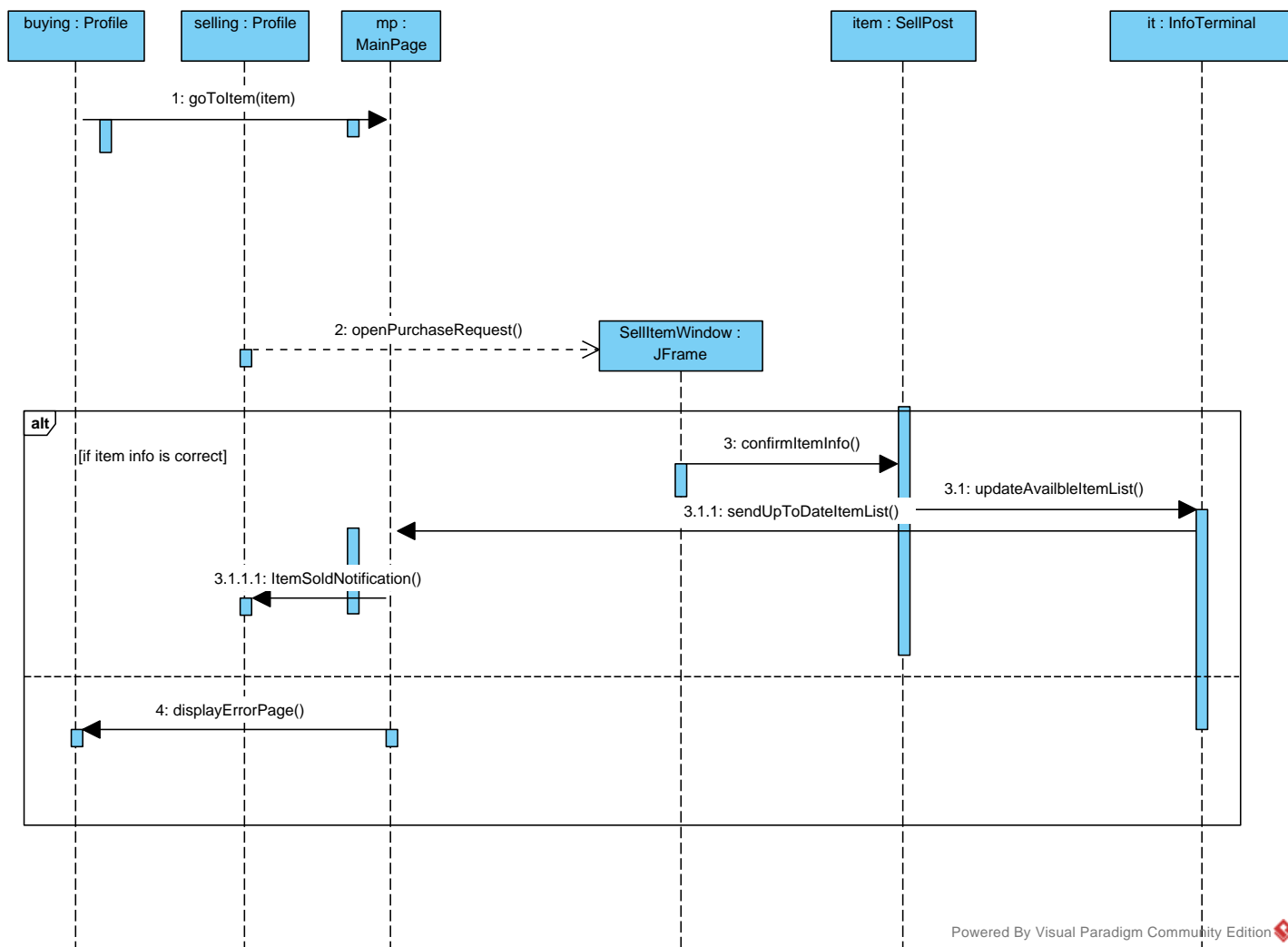












CSI3471 GroupProject

Oct 30, 2020

<http://>

Project manager

Project dates

Sep 21, 2020 - Oct 31, 2020

Completion

0%

Tasks

19

Resources

3

Tasks

Name	Begin date	End date
Iteration1	9/21/20	9/30/20
Gantt diagram set up	9/21/20	9/21/20
Website setup	9/21/20	9/21/20
Use Cases	9/21/20	9/22/20
Requirement list	9/23/20	9/23/20
Wireframes	9/23/20	9/23/20
SSD	9/24/20	9/25/20
Domain Model	9/24/20	9/25/20
System operations	9/26/20	9/26/20
Operation Contract	9/28/20	9/29/20
Presentation	9/28/20	9/30/20
Iteration2	10/24/20	10/30/20
SSD	10/24/20	10/25/20
design model	10/26/20	10/26/20
architecture	10/26/20	10/28/20
Demo of prototype	10/26/20	10/28/20
Identification of GRASP pattern	10/28/20	10/28/20
test coverage plan	10/29/20	10/29/20
update IterationII materials	10/30/20	10/30/20

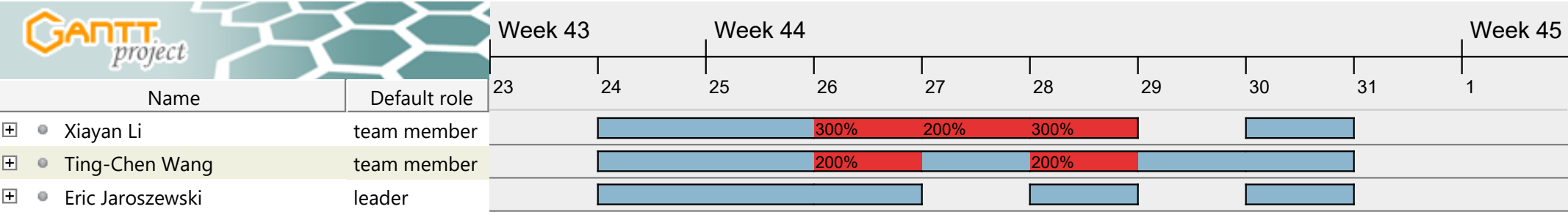
Resources

Name	Default role
Xiayan Li	team member
Ting-Chen Wang	team member
Eric Jaroszewski	leader

Gantt Chart



Resources Chart

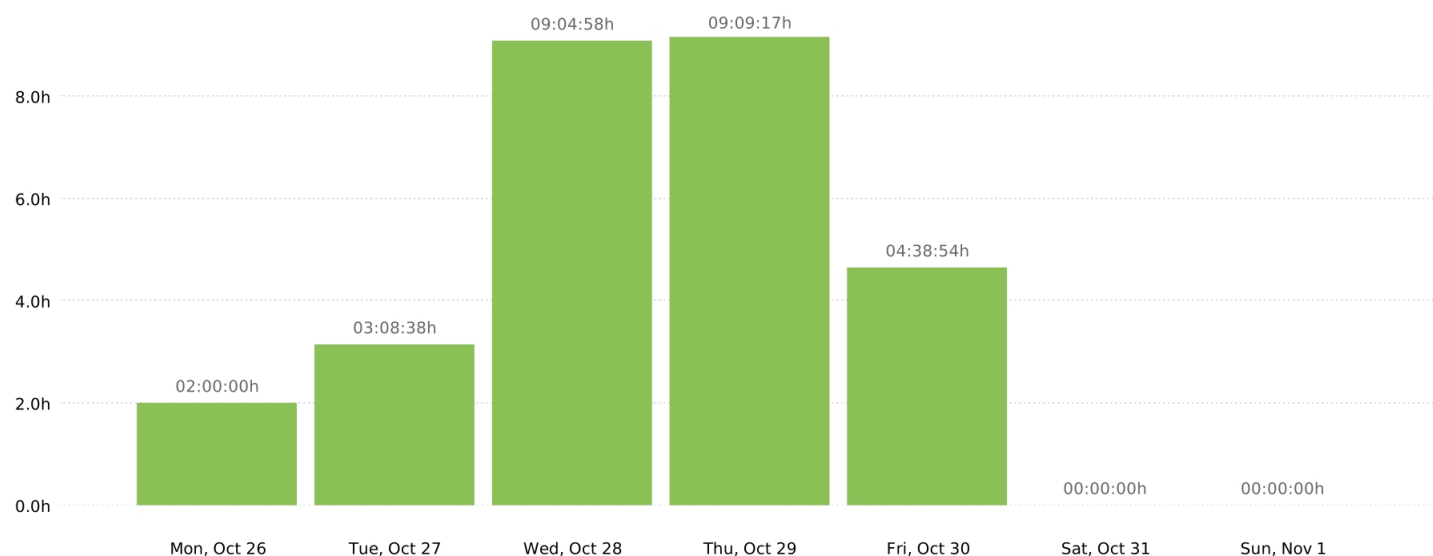


Summary report

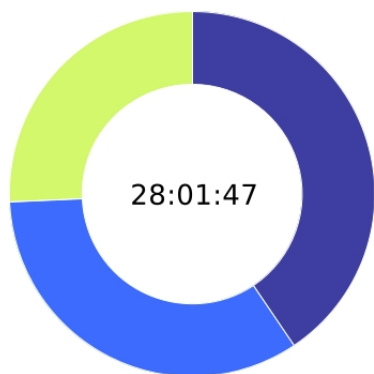


10/26/2020 - 11/01/2020

Total: 28:01:47 Billable: 28:01:47 Amount: 0.00 USD



User



Eric Jaroszewski	07:12:23	25.72%
Tina	09:27:00	33.71%
Tingchenwang2019	11:22:24	40.56%

User	Duration	Amount
Eric Jaroszewski	07:12:23	0.00 USD

Tina	09:27:00	0.00 USD
Tingchenwang2019	11:22:24	0.00 USD