

Student number	21035478
Main text body word count	1098 words

Reflective abstract (up to 150 words) Please help the marker give you useful, personally-tailored, feedback by completing the sections below. Nothing you write here will adversely affect your mark.	
What mark do you anticipate your essay will achieve?	Unsure
What do you think you did well in the writing of this essay?	I believe I drew good comparisons from the other tools I had used and effectively referenced my logbook.
What aspects of your essay do you think may need improvement?	I struggled with the smaller word count and fitting all my observations/evaluations within it.
What areas of feedback would be most helpful for you?	<ul style="list-style-type: none">Effectiveness of my references.

Introduction

The following report analyses and evaluates LucidChart as a UML modelling tool, drawing from my experiences across multiple modelling frameworks and relevant academic literature. Specifically focused on LucidChart's effectiveness in modelling the UWEFlix case study, and the following criteria:

- Cost
- Capability
- Ease-of-Use
- Visuals

In deciding upon which tools to utilise, I took inspiration from multiple contexts. These contexts included research on the most popular current tools (**Martin, 2024**), tools I was already familiar with, and tools which ranked highly in **Lu and Alexandru's (2023)** study evaluating UML tools used in higher education software engineering courses. The modelling frameworks I decided for comparisons were draw.io, AstahUML and Microsoft Visio. This allowed me to get an eclectic mix, encouraging a more thorough evaluation as their selection was derived from multiple sources, theoretically giving me a wider range of knowledge to draw comparisons to my main tool. I selected LucidChart as my main tool due to its popularity (**Martin, 2024**) and high ranking (**Lu and Alexandru, 2023**).

Evaluation Criteria

The first of my evaluation criteria is cost, with focus on the tool's value for money and whether it operates on a paid or free model. In an industry context this criterion is essential, cost is a tangible and measurable factor that can be combined with other criteria to provide cost-benefit analysis (**Investopedia, 2024**), helping identify whether the expense improves output to an acceptable degree.

My second criteria is capability, this covers a wide range of attributes, one key consideration is the tool's range of diagram templates/assets and their relevance, which can be extended to also consider the tool's efficiency when working outside of pre-defined assets. Another consideration is an evaluation of the tool's AI features, primarily due to its increasing relevance in assisting UML modelling, **Cámara et al. (2023)** believe AI is "destined to play a prominent role in modelling". My final consideration is collaborative capabilities, almost 86% of software developers work in agile teams (**Unosquare, 2024**) requiring collaboration, for large enterprise systems it is highly unlikely you will be the only person creating diagrams, making collaborative capabilities highly important.

My third criteria is ease-of-use, it links to cost-benefit analysis and capability in that it helps gauge efficiency of diagram creation, simultaneously evaluating the tool's level of output. Ease-of-use evaluates the learning curve of using the tool, in industry a complex tool can require training/research, which does have its advantages, but slows down output and can be costly (**Grzegorzek, J., 2023**). I will be evaluating whether the tool is easy to use relative

to the complexity of its capabilities, to prevent unfair evaluation for having extensive features.

My final criteria, visuals, considers the appearance of the diagrams the tool produces and their styling. My reasoning for this is to gauge the legibility of the diagrams and identify possible accessibility concerns, a tool with verbose options for styling will be stronger at following UML aesthetic guidelines such as those presented by **Eichelberger and Schmid (2009)**. In presentations to stakeholders/managers within industry, effectively styled diagrams draw attention to important features (**SmartDraw, 2024**), enhancing their impact and highlighting visual importance for an effective UML tool.

Evidence and Analysis

LucidChart is a freemium web app (**Log 4**), its free version limits available assets, the maximum number of assets in a single diagram, and the number of editable files a user can create. For a large project the free version would not be conceivable, its limitations are too great to justify (**Log 8 & Appendix B, Figure 1**). Comparatively, draw.io (**Log 7**) is entirely free and offers enough features for use in large projects, partially shown in the complexity of **appendix C, figures 1 & 4**. LucidChart's paid model, however, offers a polished product that gives users additional capabilities. These capabilities include AI assistance, external libraries, in-depth collaborative tools, and extensive styling. As discussed in **log 5 & 8**, I found these features endlessly useful, demonstrated best visually in **appendix B, figures 2, 5 & 6**. LucidChart's paid model offers vast capabilities, though draw.io comes impressively close as a free alternative (**Log 7**). This sparks considerations of LucidChart's value for money, I personally found the improved product justified the additional cost.

To gauge the level of these improvements, I created my interface diagram in both LucidChart and Visio (**Appendix B, Figure 2 & Appendix C, Figure 2**). Comparatively, using Visio was very time-consuming, the result was also less visually impressive than the LucidChart alternative (**Log 5**). This highlighted LucidChart's capability for efficiently creating diagrams without templates, all whilst remaining visually impressive. Despite starting from scratch, modelling was efficient due to features such as AI assistance, which helped create custom shapes from text prompts.

Ease-of-use is enhanced by LucidChart's drag-and-drop interface showing exactly where an asset will be placed, this felt more intuitive than for example, AstahUML (**Log 6**), and allowed me to effectively layout diagrams such as **appendix B, figure 3**. It also made following **Eichelberger and Schmid's (2009)** UML guidelines much easier. LucidChart's setup is straightforward but requires an account and internet connection (**log 4**), whilst AstahUML has offline support (**Log 6**), and draw.io requires zero setup. Accentuating the fact tools have their own pros and cons, including LucidChart. After initial setup, LucidChart's cloud-architecture ensured it required minimal hardware requirements, and an account's files/diagrams were accessible from most devices via web browser, promoting its ease-of-use over its competitors (**Log 4**).

LucidChart's extensive styling options allows users to present a desired aesthetic, with themes promoting consistencies across diagrams (**Log 8 & Appendix B**). Effectiveness of

styling in LucidChart is highlighted in the contrast between my **Appendix C** diagrams and their LucidChart alternatives, LucidChart diagrams are much more eye-catching whilst remaining visually consistent.

The lack of cross-diagram links between assets improves ease-of-use but decreases efficiency when you have many diagrams using the same assets, as is the case with the UWEFlx case study (**Appendix B**). This feature is in AstahUML (**Log 6**), LucidChart could benefit from its implementation considering large-scale system design often requires many diagrams.

Conclusions

LucidChart's paid model offers vast styling options, fantastic ease-of-use and plenty of capabilities. This helps it to excel in agile teams and enterprise systems, including use at DocuSign, Delta Airlines, Hitachi and Western Union (**LucidChart, 2024**). Although lacking cross-diagram references and offline functionality, LucidChart efficiently modelled diagrams for the UWEFlx case study, and a large amount of its features went unused. It's high scores in UML tool analysis (**Figure 1**) provides further support for its competency as a modelling framework. LucidChart also scored highest in my personal scoring of the tools I used against my evaluation criteria (**Figure 2**).

Tool Name	Edraw [6]	Visual Paradigm [21]	MagicDraw [14]	Gliffy [10]	Lucidchart [13]	Creately [4]	UMLetino [20]	Draw.io [5]	GitMind [9]	SmartDraw [19]
Use Case Diagram (out of 10)	10	10	8	9	9	9	9	7	6	8
Class Diagram (out of 47)	37	30	33	36	34	28	40	42	36	31
Sequence Diagram (out of 41)	34	32	34	36	35	29	27	36	30	25
State Diagram (out of 22)	15	12	22	17	11	15	16	14	11	14
Activity Diagram (out of 18)	12	14	16	14	6	13	16	10	10	13
Dedicated support for diagram types (out of 5)	5	5	5	5	3	5	5	0	3	5
Export functionalities (out of 6)	4	4	3	2	3	3	2	5	3	0
Collaboration Functionalities (out of 16)	14	15	0	1	15	13	0	0	1	1
TOTAL SCORE (out of 165)	131	122	121	120	116	115	115	114	100	97
Tool Name	Gaphor [7]	Cacoo [2]	Moqups [16]	Sketchboard [18]	OmniGraffle [17]	GenMyModel [8]	ConceptDraw [3]	yEd [22]	Microsoft Visio [15]	Archi [1]
Use Case Diagram (out of 10)	8	8	9	5	8	8	3	7	8	2
Class Diagram (out of 47)	24	30	32	24	33	36	21	12	0	5
Sequence Diagram (out of 41)	26	31	20	8	12	18	0	2	0	0
State Diagram (out of 22)	18	11	11	12	11	0	11	4	10	2
Activity Diagram (out of 18)	12	10	5	12	5	2	2	1	2	1
Dedicated support for diagram types (out of 5)	3	0	1	0	3	2	0	0	0	0
Export functionalities (out of 6)	3	2	0	3	0	2	0	0	0	0
Collaboration Functionalities (out of 16)	0	1	14	14	0	2	0	0	0	0
TOTAL SCORE (out of 165)	94	93	92	78	72	70	37	26	20	10

Figure 1 Total scores for various UML modelling tools (Lu and Alexandru, 2023)

UML Tool	Cost	Capability	Ease-of-Use	Visuals	Total
LucidChart Paid	4/5	4.5/5	4.5/5	5/5	18
LucidChart Free	5/5	1/5	4/5	4/5	14
Draw.io	5/5	4/5	4.5/5	3/5	16.5
AstahUML	3/5	4/5	2/5	4/5	13
Microsoft Visio	3/5	3/5	4/5	2/5	12

Figure 2 Personal Evaluation of Tools Used

References

- Cámara, J., Troya, J., Burgueño, L. and Vallecillo, A. (2023) On the Assessment of Generative AI in Modelling Tasks: An Experience Report with ChatGPT and UML. **Software and Systems Modelling** [online]. 22 pp. 781-793. [Accessed 27 November 2024]
- Eichelberger, H. and Schmid, K. (2009) Guidelines on the Aesthetic Quality of UML Class Diagrams. **Information and Software Technology** [online]. 51(12) pp. 1686-1698. [Accessed 27 November 2024]
- Grzegorzek, J. (2023) Advantages and Disadvantages of Employee Training. **Super Business Manager** [blog]. 18 December. Available from: <https://www.superbusinessmanager.com/advantages-and-disadvantages-of-employee-training/> [Accessed 27 November 2024]
- Investopedia (2024) **Cost-Benefit Analysis**. Available from: <https://www.investopedia.com/terms/c/cost-benefitanalysis.asp> [Accessed 27 November 2024]
- Lu, Y. and Alexandru, A.C. (2023) Systematic Review of UML Diagramming Software Tools for Higher Education Software Engineering Courses. **Proceedings of the 2023 Conference on United Kingdom & Ireland Computing Education Research** [online]. Swansea, Wales, UK, 25 September 2023. UKICER, pp. 1-7. Available from: <https://dl.acm.org/doi/10.1145/3610969.3611123> [Accessed 25 November 2024]
- LucidChart (2024) **How 4 Companies Have Used LucidChart to Improve Their Processes**. Available from: <https://www.lucidchart.com/blog/how-4-companies-have-used-lucidchart-to-improve-their-processes> [Accessed 27 November 2024]
- Martin, M. (2024) The Best Flowchart Software and Diagram Tools in 2025. **Zapier** [blog]. 5 November. Available from: <https://zapier.com/blog/flowchart-diagramming-software/> [Accessed 29 November 2024]
- SmartDraw (2024) **Tips for Effective UML Diagrams**. Available from: <https://www.smartdraw.com/uml-diagram/uml-diagram-tips.htm> [Accessed 27 November 2024]
- Unosquare (2024) **Agile Development 101**. Available from: <https://www.unosquare.com/blog/agile-development-101/> [Accessed 27 November 2024]

Appendix (A)

For the sake of clarity, I have uploaded all 8 of my logs, as the later logs reference the earlier ones. However, the logs I have chosen for my best 5 logs are:

- 4
- 5
- 6
- 7
- 8

Diagrams attached to my logs are drafts from the week they were created, final diagrams are in Appendix B & C.

Log 1

Student's	
ID: 21035478	Name: Arthur Milner
Team members: James, Bradley	
Date: 04/10/24	Subject: Introduction into UWEFlix Case
<p>Critical Evaluation:</p> <p>We worked well as a group to identify stakeholders, potential technologies and a rough development plan. Although I did critically analyse the other members' points, for the most part I was focused on supporting and expanding the ideas presented to me. Due to the idea of an enterprise system being new for the group, it was hard to imagine how to begin in considering the steps required to build one, despite this we were able to brainstorm some good ideas together.</p> <p>Hopefully in the future this will be easier to picture as I get a better idea of the development timeline for an enterprise system and give more depth to our analysis of building enterprise systems. The session helped me to identify with my group pre-existing resources/frameworks you might use in developing an enterprise system, such as Django, Flask, SQL, Docker, external payment systems, etc. Many ideas were things I would have struggled to find by myself. At the end of the session, I felt I had a much better understanding of the case study, it helped to break the problem down with the identification of stakeholders and which of those are users.</p>	
Approved by: James Vickers Date: 11/10/24	

Log 2

Student's	
ID: 21035478	Name: Arthur Milner
Team members: James, Amin, Matt, Cosmo, Krishna	
Date: 11/10/24	Subject: Lifecycles for Large Scale Software Development

Critical Evaluation:

I personally possessed minimal experience in the concept of SCRUM/agile development, this required me to listen to the knowledge from my group members and do my own research, consequently, I didn't contribute as much as my peers to group discussion.

As a group we found ourselves often running out of time before completing a task to a standard we were happy with, considering future group work I would suggest splitting tasks out between members. For example, when tackling task 2, one member research scrum master, one member research product owner, etc.

The size of the group was also a challenge with many ideas coming in at the same time, it almost felt as if we were brainstorming for the tasks rather than crafting an answer with more depth. We had two members join the session late which halted our progress slightly as we had to catch them up with what we had already discussed, luckily the notes we took from our conversations before their arrival were detailed enough to minimise the delay. As a group we also adapted well to faulty equipment which made the whiteboard inaccessible and moved to a word document.

Approved by: James Vickers

Date: 18/10/24

Log 3

Student's

ID: 21035478

Name: Arthur Milner

Team members: Rohaan, Krishna, Ana, Dongyiu, Ugo, Shurida

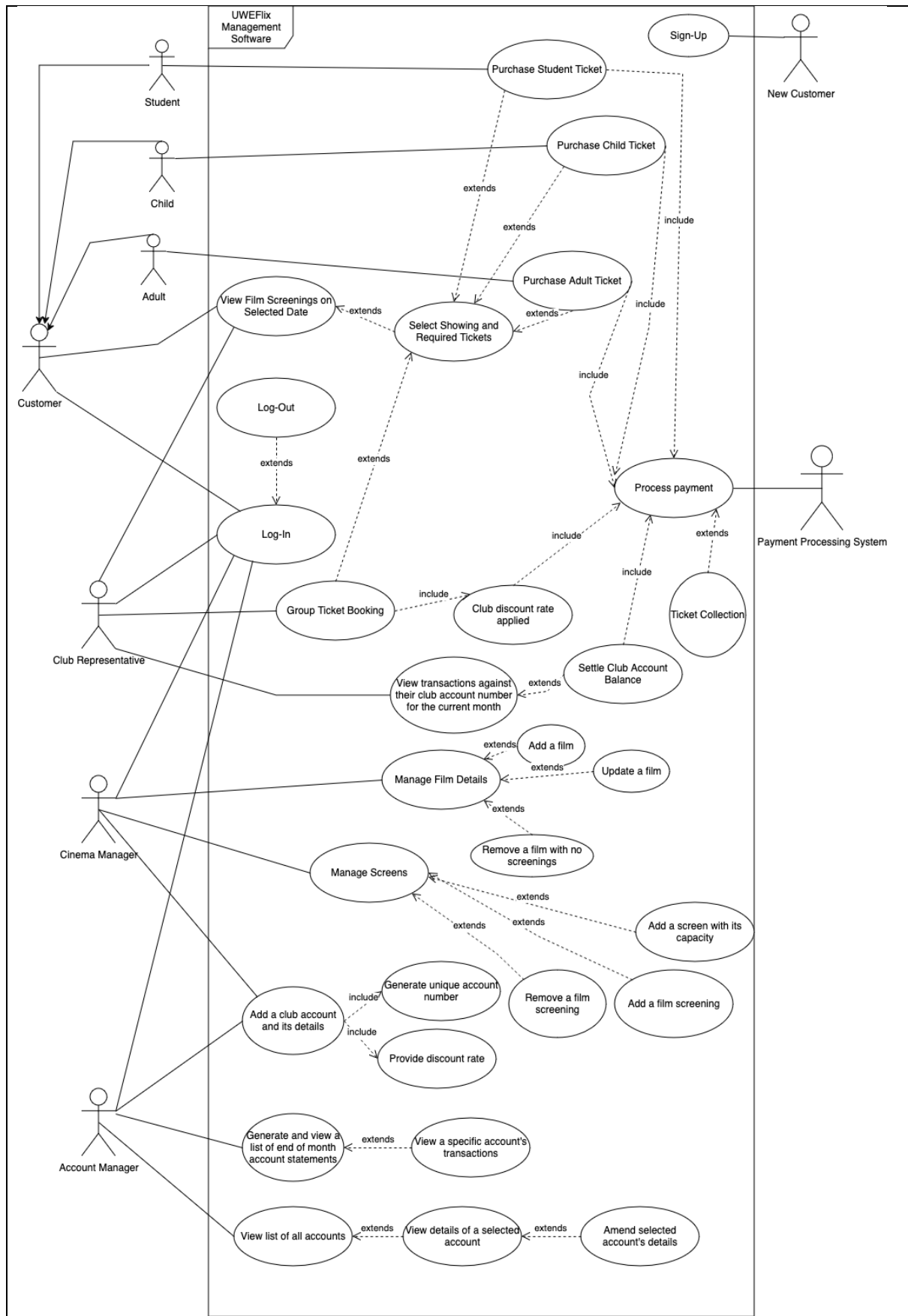
Date: 18/10/24

Subject: Requirements Handling

Critical Evaluation:

My group were able to finish the tasks in a timely manner by focusing down the problem into smaller chunks. For example, identifying the requirements per user before creating a larger list of requirements. There were some members of my group who weren't contributing a lot to group discussion, it was challenging to involve them and get their insight into the problem, attempting probing questions in the future might encourage involvement. As a team we also used feedback to refine our ideas for our presentation, although in the future I would try asking for feedback more frequently to prevent investing time in ineffective ideas.

Independently I used draw.io to create my diagram, work done in session was the basis for this. I could've been more thorough in my requirement identification, there were clear requirements that were missed in the session. Draw.io being web based made set-up easy, but I feel as though it was lacking the aesthetics of similar tools and found the interface quite difficult when positioning objects. Layout was the main challenge I encountered for this task, moving forward I will take more care in planning layout before starting creation.



Approved by: James Vickers
Date: 25/10/24

Log 4

Student's

ID: 21035478

Name: Arthur Milner

Team members: Rohaan,
Amin, Shabhaz, Nathan

Date: 25/10/24

Subject: Non-Functional Requirements/Interface Diagram

Critical Evaluation:

Initially, we struggled to combine our thoughts as a group, having different functional requirements from previous weeks required time to agree on a definitive list of requirements. We had issues with space on the whiteboard, failing to leave room to discuss test cases. Consequently, the legibility and ability to present our work suffered. We gave a comprehensive answer to the tasks, but we struggled to discern interface types efficiently and it required some reflection.

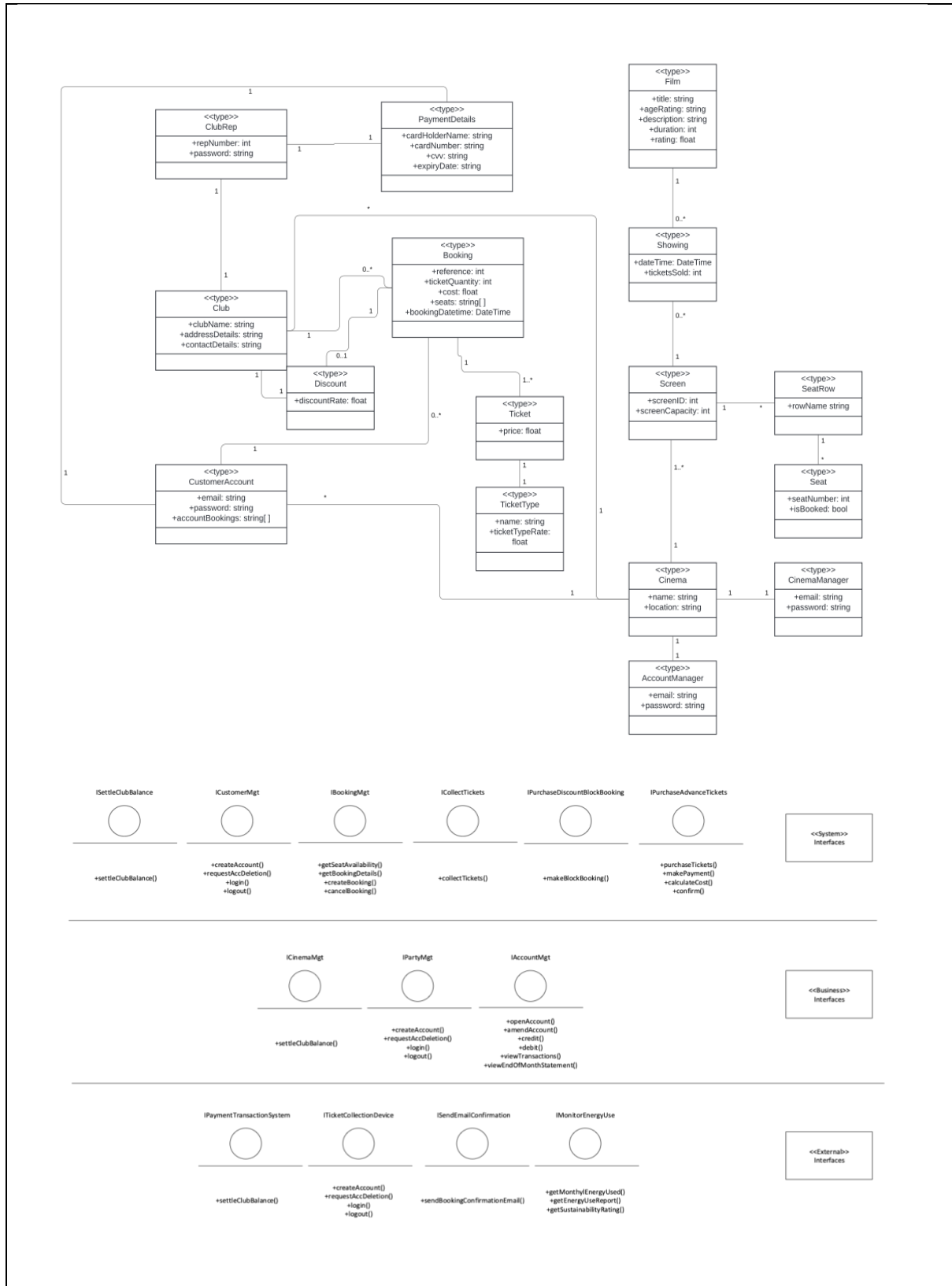
I used LucidChart to design my interface diagram, the tool shared the draw.io convenience of being web based but required account registration. Certain features require a paid license, such as a high number of assets in a single file, here free content was sufficient, but this could prove problematic for more complex diagrams. LucidChart autosaved progress as opposed to draw.io which gives peace of mind and minimises potential for lost work. I found the export process seamless in LucidChart, you can easily crop and export diagrams using the GUI. LucidChart saves diagrams to your account, which is not possible within draw.io. LucidChart did not have the pre-built assets for unconnected interfaces, so I have used classes for now and will refine this next week.

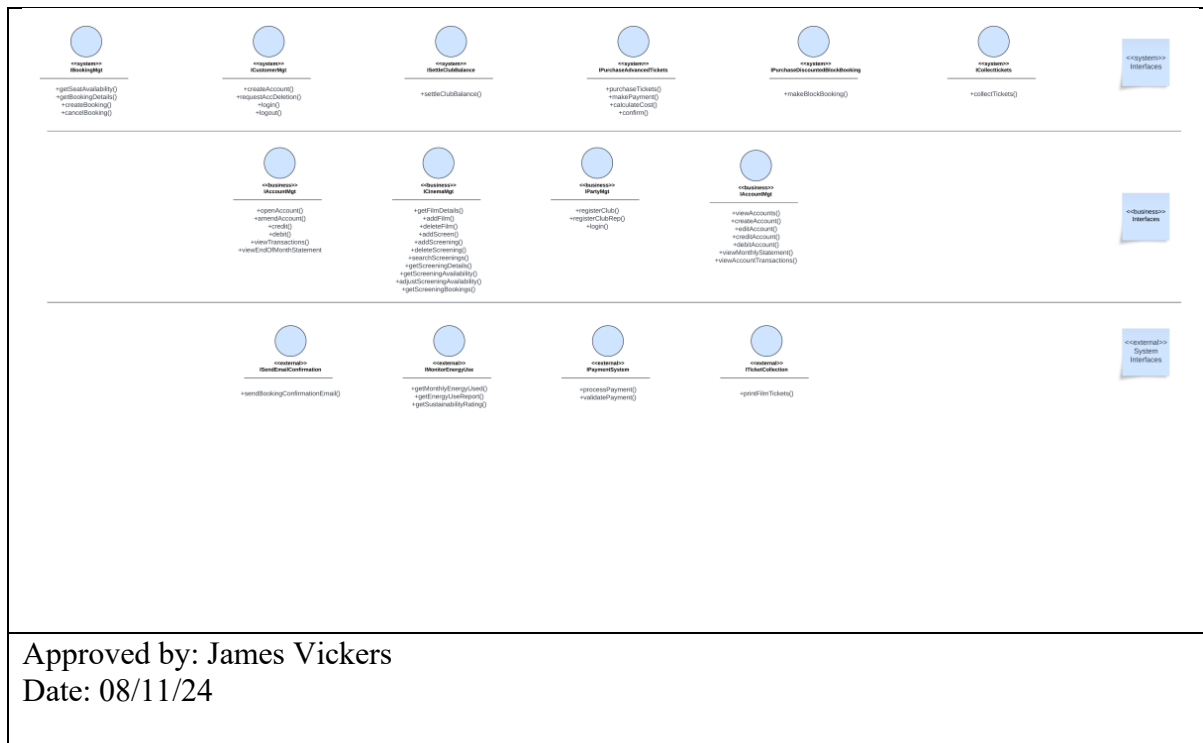


Approved by: James Vickers
Date: 01/11/24

Log 5

Student's	
ID: 21035478	Name: Arthur Milner
Team members: Rohaan, William, Troy, Bradley	
Date: 01/11/24	Subject: Interface Diagram/Components & Services
<p>Critical Evaluation:</p> <p>This week I thought we worked very well as a group. We used agile methodology to uncover refining details of last week's interface diagram. I would like to take some of what made us work well into future groups/weeks, for instance having a SCRUM leader delegate tasks, and brainstorming that built upon other's ideas. There was a disagreement about who should be SCRUM leader, which was efficiently resolved through a vote. I also took inspiration from other group presentations when improving my interface diagram.</p> <p>Following effective group work I endeavoured to fine-tune last week's interface diagram and create a business type diagram. For the business type diagram, I used LucidChart, whilst they didn't specifically have a business type diagram template, I was able to modify a class diagram for a similar effect. Knowing the limits of LucidChart's assets, I adapted the tool's other shapes to reformat my interface diagram. Creating the diagram from scratch was more time consuming than using pre-made assets, however, I attempted the very same in Visio to provide a basis of comparison and found LucidChart much more intuitive due to its verbose capabilities and assets. Visio also lacked the visual flair of its LucidChart alternative.</p>	

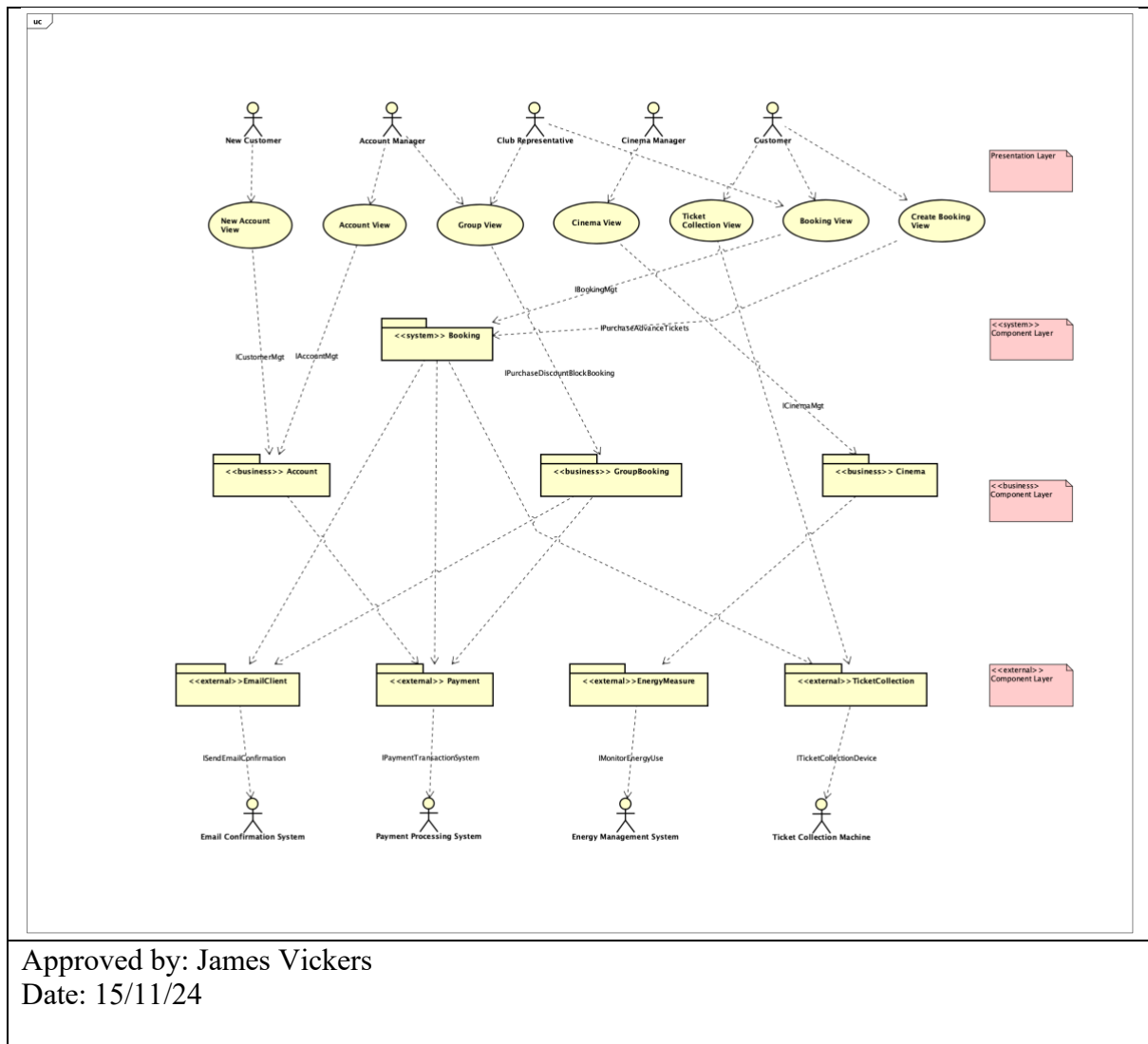




Approved by: James Vickers
Date: 08/11/24

Log 6

Student's	
ID: 21035478	Name: Arthur Milner
Team members: Jack, Jack, Ana, Iyede, Namdeen	
Date: 08/11/24	Subject: Interfaces & Components (Part 1)
<p>Critical Evaluation:</p> <p>Our group struggled with identifying components, originally, we created components that included verbose detail of the types inside. Due to time constraints, we decided to change to a higher level analysis to capture a more comprehensive picture of the system's components. We combined types from our previous diagrams into the components, doing so helped members discover new details they hadn't previously thought of. Effective time management allowed us touch on the topics for next week's session, the agile approach of members having roles once again proved effective.</p> <p>AstahUML has a steep learning curve and requires installation alongside a paid subscription, making it difficult to setup and limiting ease-of-use. I see the benefits of AstahUML for large-scale projects, it has collaborative features and nicely groups diagrams into a single .asta file, albeit these collaborative features are not as responsive as the real-time cloud-based collaboration of LucidChart. Being limited to assets in your selected diagram type was a hinderance as they didn't have an option for component architecture diagram, I adapted the assets for a use case diagram to overcome this. I found AstahUML to be priced reasonably for a tool with an expansive feature set and a USP of offline functionality.</p>	



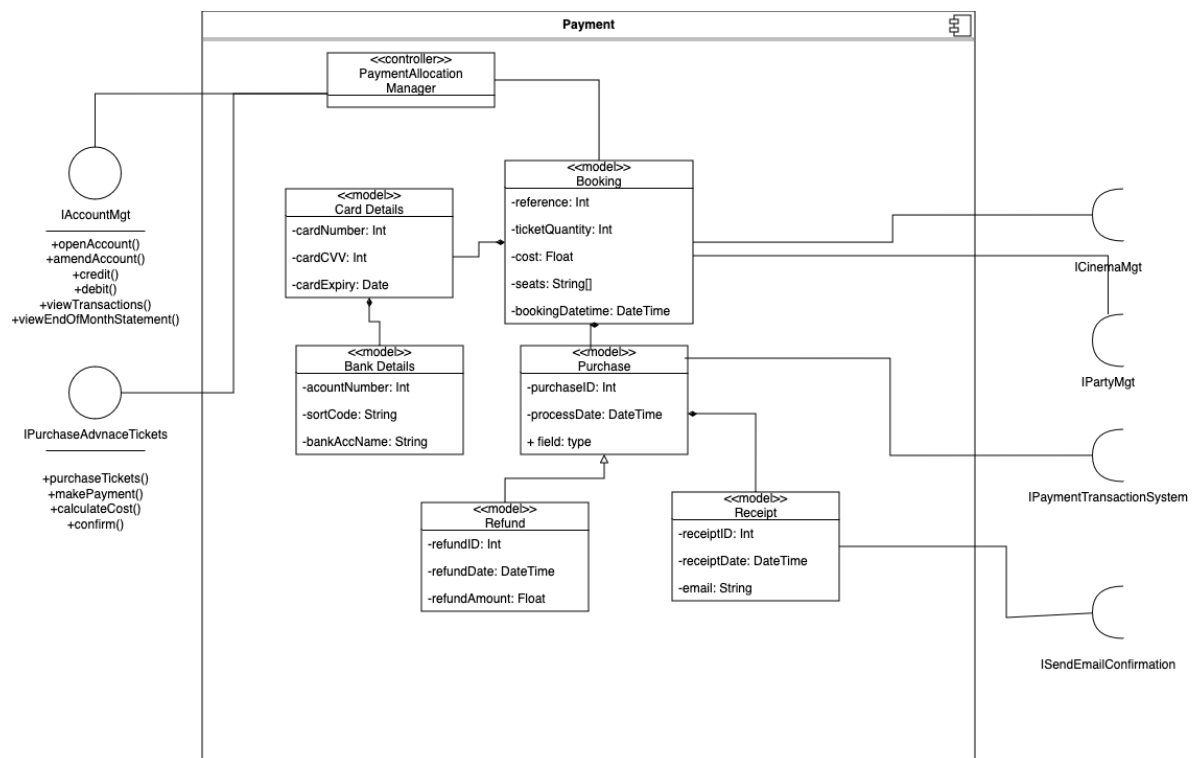
Log 7

Student's	
ID: 21035478	Name: Arthur Milner
Team members: N/A	
Date: 15/11/24	Subject: Interfaces & Components (Part 2)

Critical Evaluation:

In the session being able to compare and give/receive feedback on my own and my peers' work was a great help in directing my design after the session, consequently, believe I utilised peer feedback well. I did, however, find it hard to critique others diagrams as it was difficult to find things incorrect with their design, having minimal experience myself.

I focused on creating the payment component specification, this week I returned to draw.io. I wanted to compare its effectiveness with the tools I am now familiar with. In my original critique of draw.io I feel I was somewhat harsh, perhaps the high volume of assets within my use case was what led me to conclude it as clunky. I found the tool much easier this time around and got familiar with inputting number values for the length/width of assets to get precise accuracy. Coming from using AstahUML this was a nice change as draw.io is entirely free and a very beginner friendly tool. Draw.io includes similar cloud collaboration to LucidChart, the capabilities are superior to LucidChart's free model as it has no restrictions but inferior to LucidChart's paid model as it lacks things such as AI and advanced collaboration.



Approved by: James Vickers

Date: 22/11/24

Log 8

Student's	
ID: 21035478	Name: Arthur Milner
Team members: Nathan, Nour, Olanrewaju, Rohaan, Ugo	
Date: 22/11/24	Subject: Sequence Diagrams

Critical Evaluation:

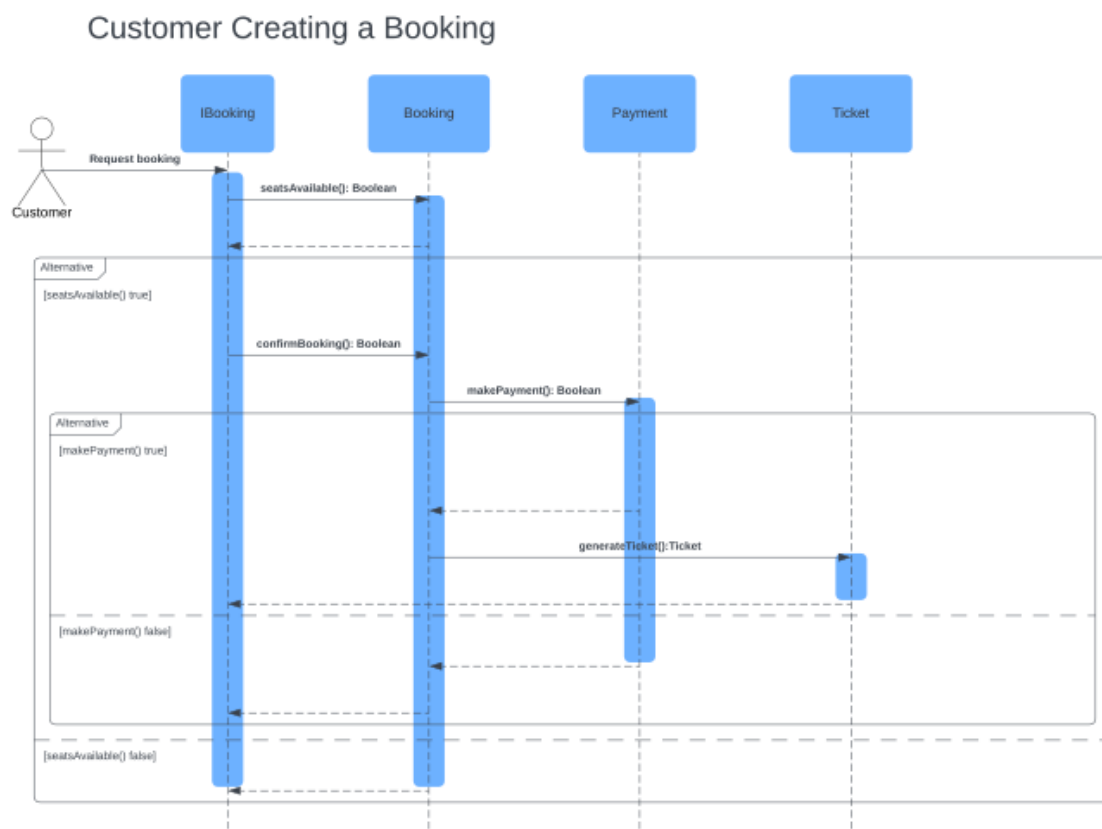
Group work and tutor feedback, inspired the creation of a sequence diagram, alongside previous diagrams I had not yet created with LucidChart.

In the creation of this diagram I wanted to use functionalities I hadn't used much, for example the AI assistant. I found the assistant generated an okay diagram, but missed key details, such as alternatives and labels. It was useful for quickly drafting up a diagram and generating custom shapes, but would fall short without quality checks, it is also possible this was due to myself sending sub-par text prompts.

LucidChart also has a marketplace of extensions which greatly expands its functionality over its peers, such as connections with AWS/Azure, very popular tools in industry. LucidChart is superior collaboratively compared to other tools I have used, its cloud-based real-time collaboration has options for annotations, and has the capability to host a meeting.

I had to use a free trial as the limits of LucidChart free were too great to create all my diagrams, namely the file and asset limit.

LucidChart has great options for styling, including conditional formatting and a selection of themes, this ensured consistency and improved the overall aesthetics my diagrams.



Approved by: James Vickers

Date: 29/11/24

Appendix (B)

Appendix of final LucidChart diagrams following weekly feedback/adjustments over time.

Figure 1 (Use Case Final)

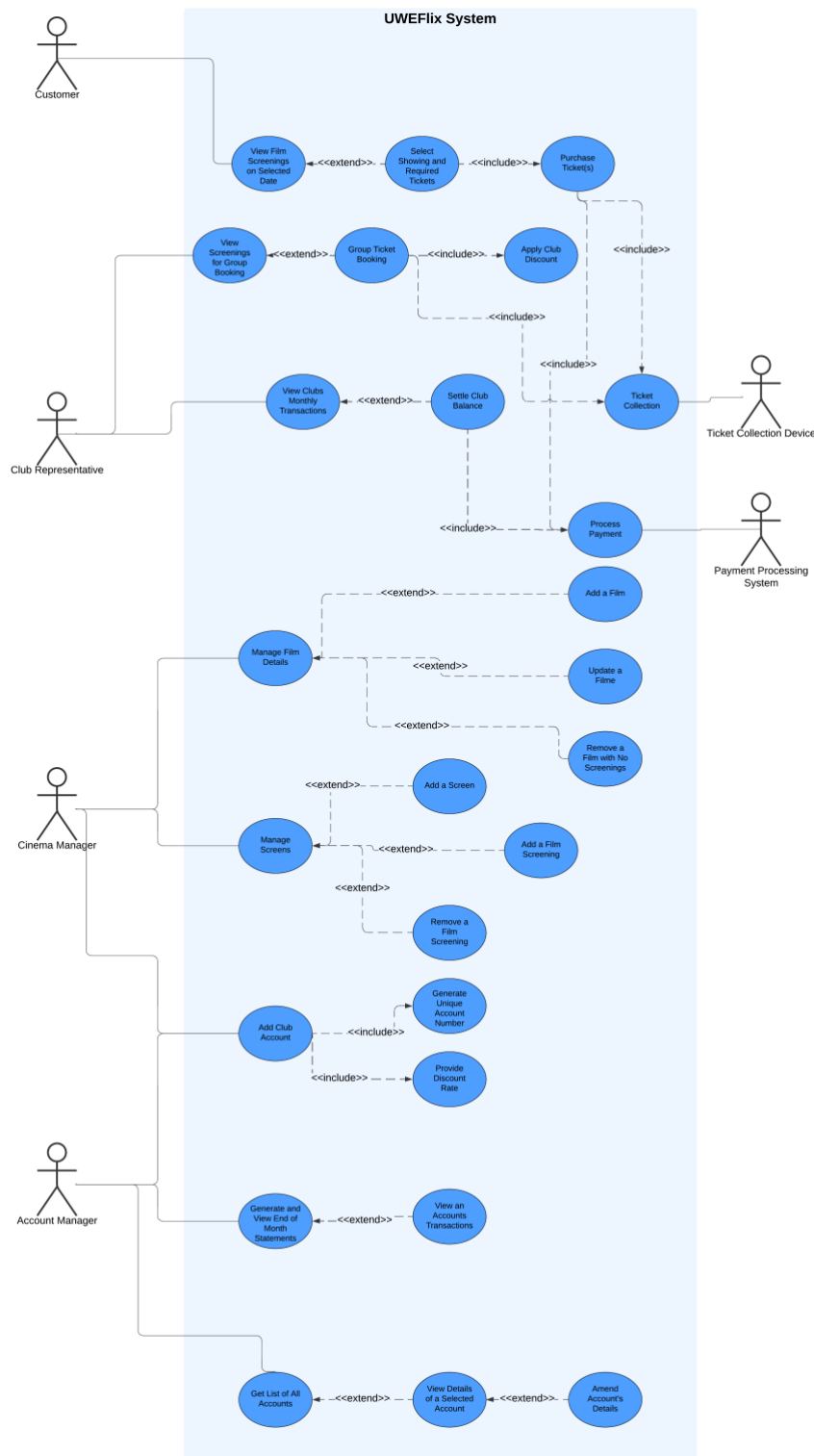


Figure 2 (Interface Diagram Final)



Figure 3 (Business Type Diagram Final)

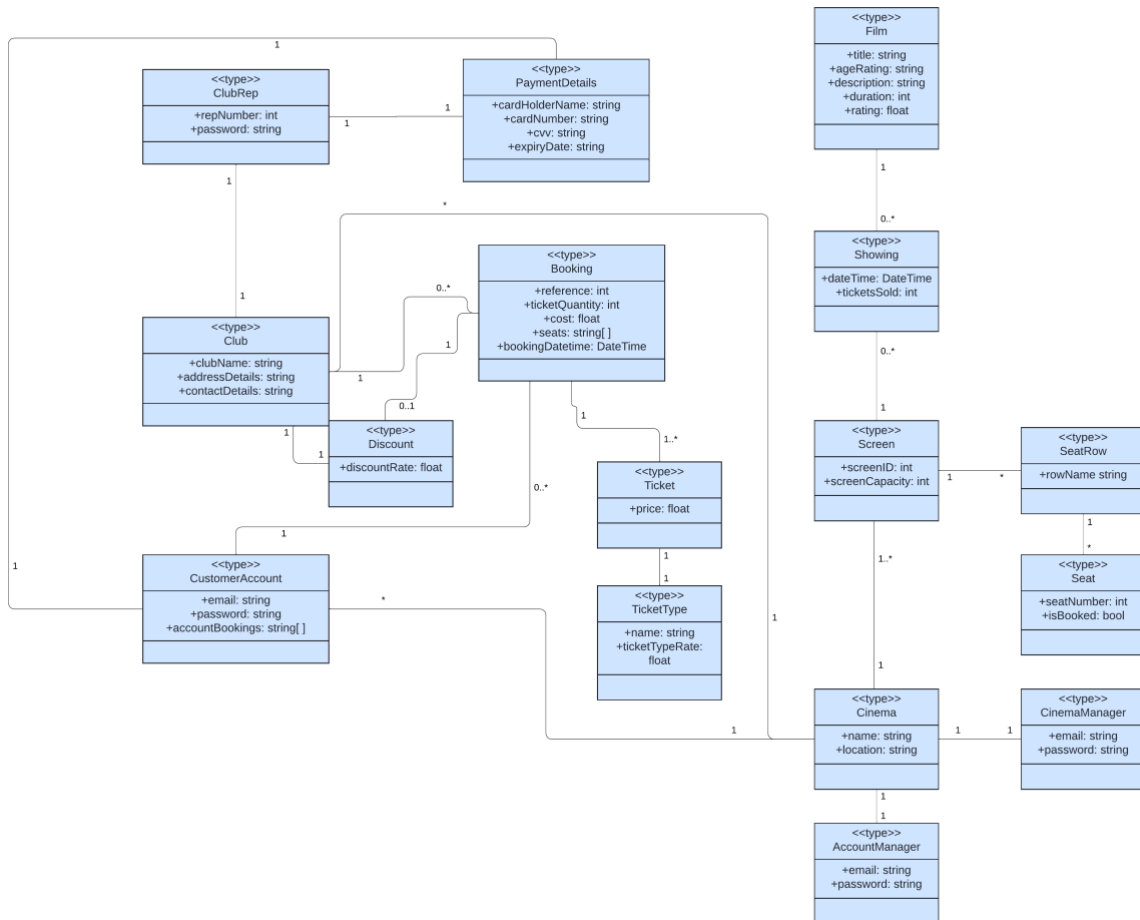


Figure 4 (Component Architecture Final)

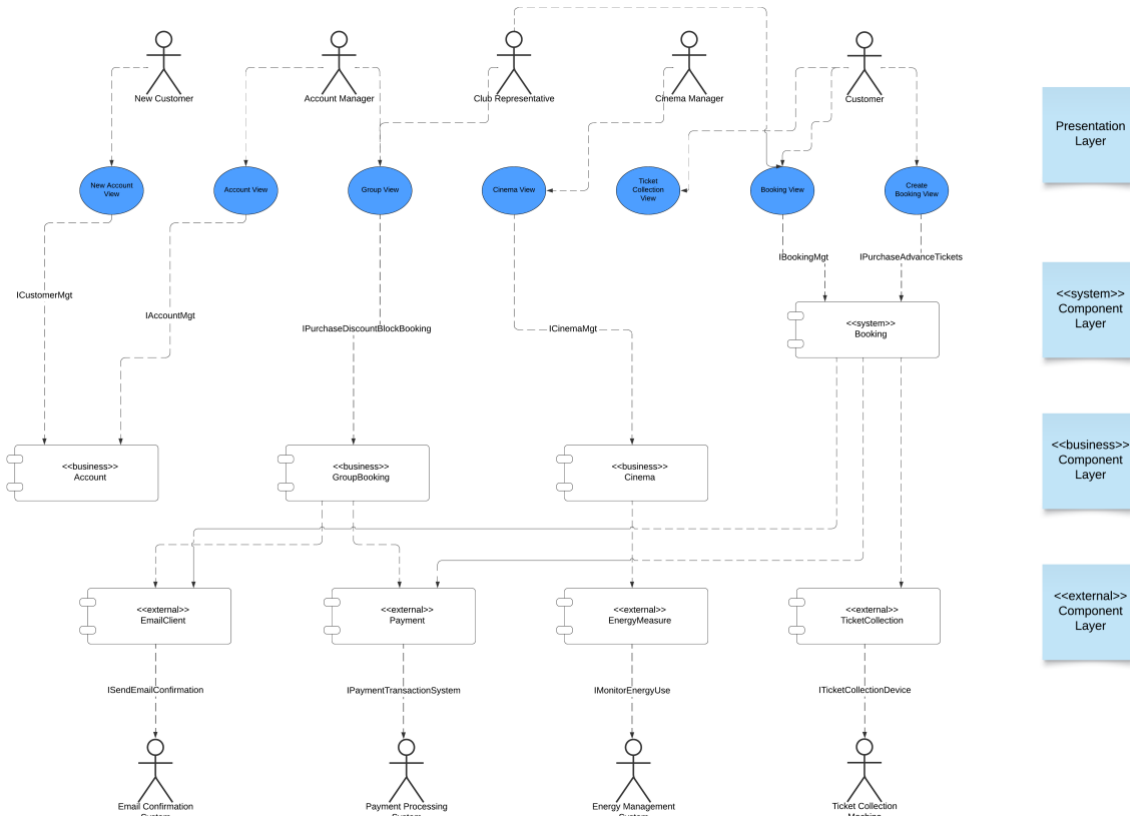


Figure 5 (Component Specification Final)

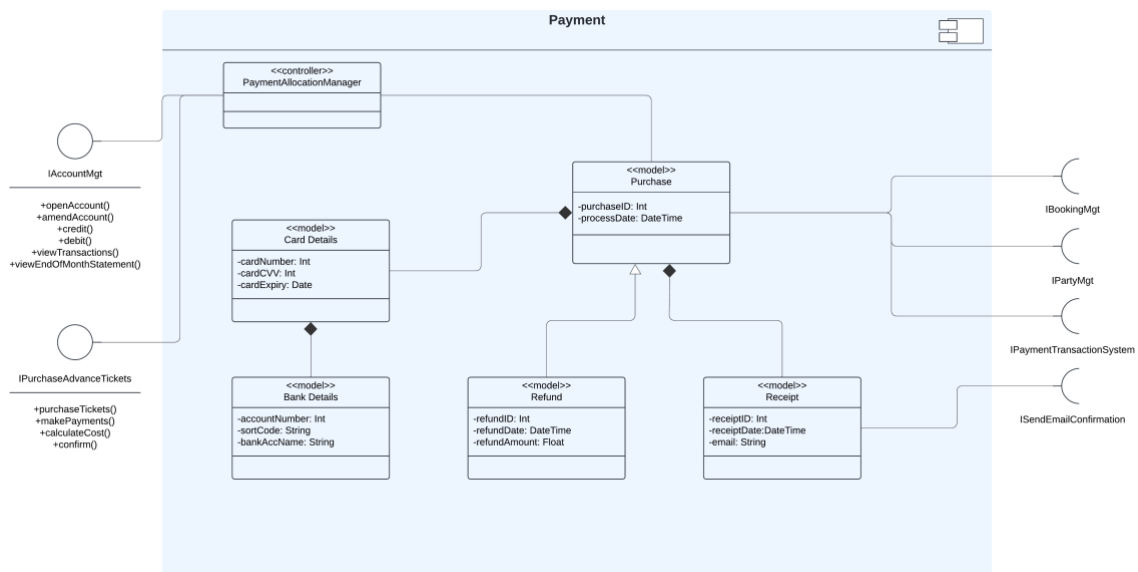
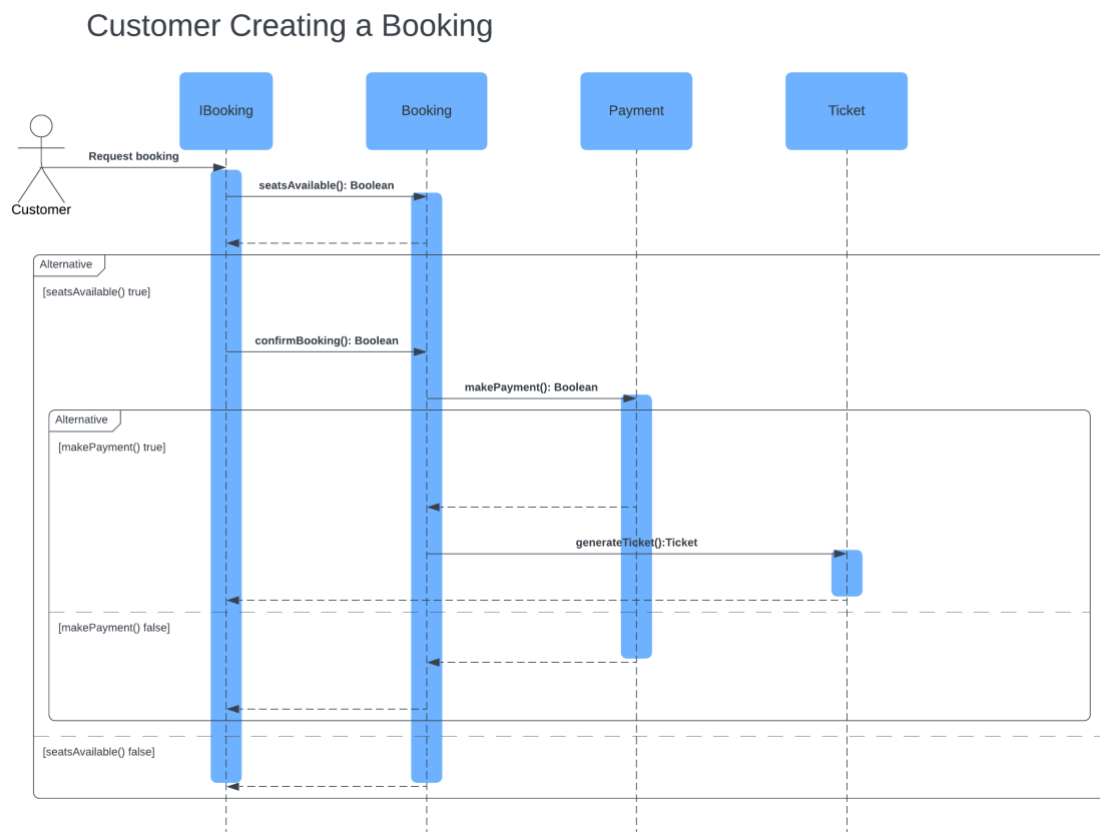


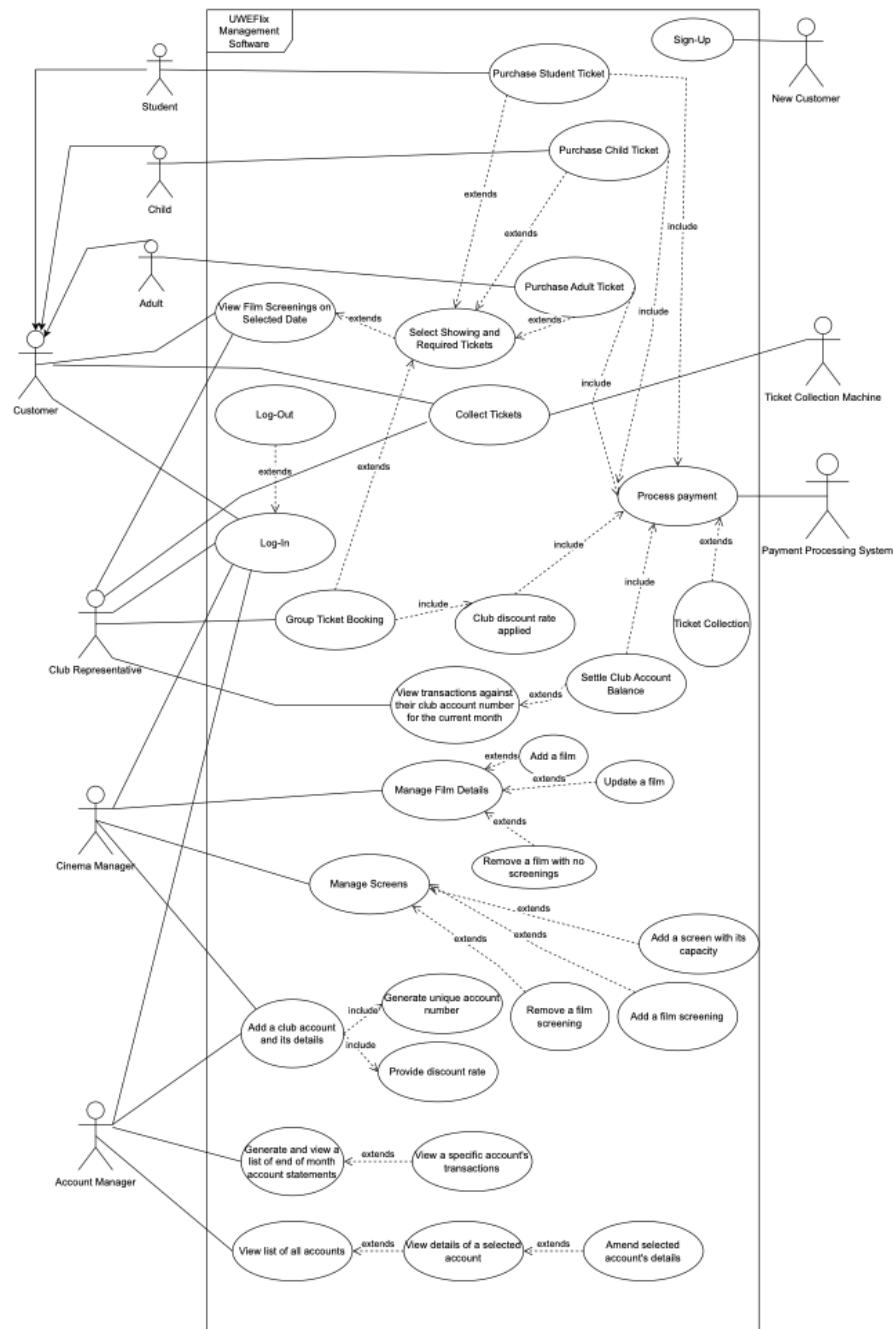
Figure 6 (Sequence Diagram Final)



Appendix (C)

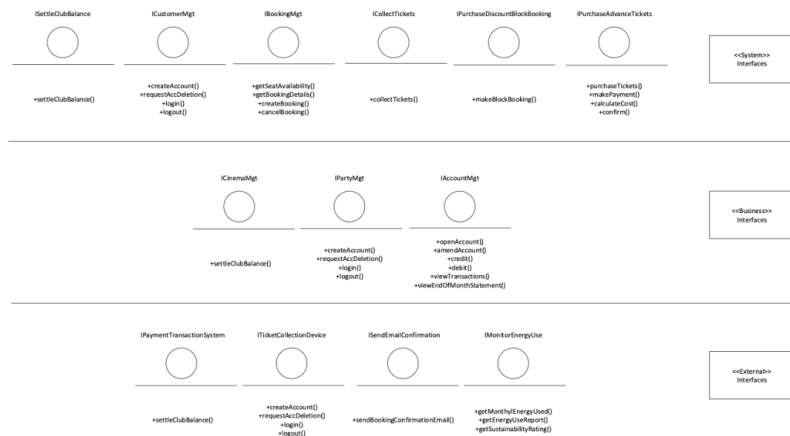
Appendix of the diagrams I created in tools I used for comparisons. Tool which was used in each diagram is captioned.

Figure 1 (Use Case)



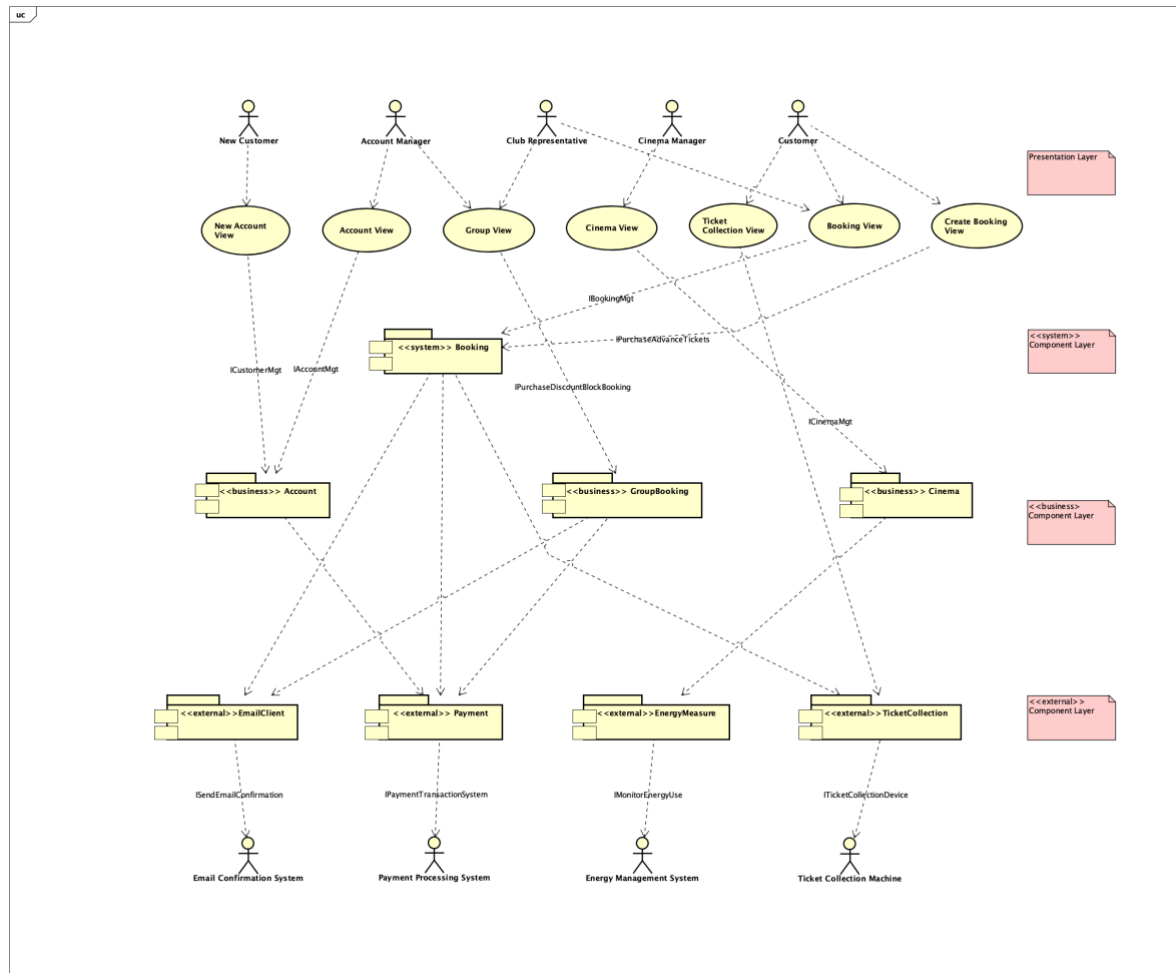
Tool Used: Draw.io

Figure 2 (Interface Diagram)



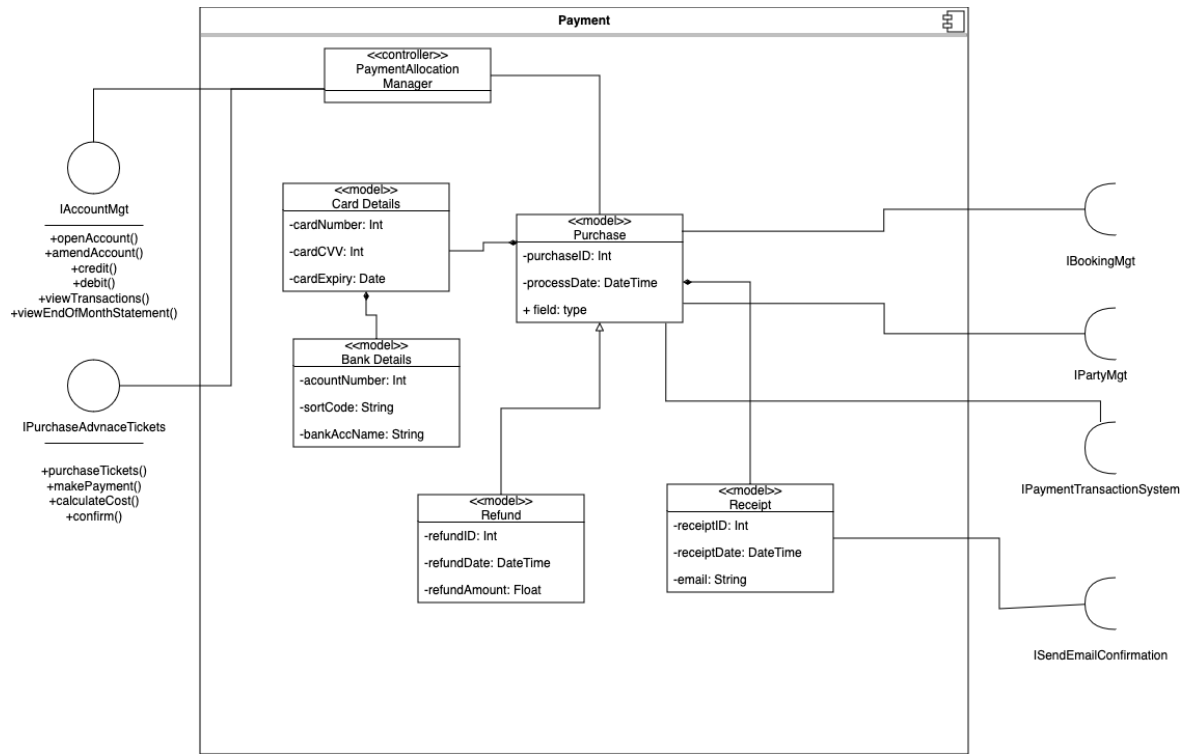
Tool Used: Microsoft Visio

Figure 3 (Component Architecture)



Tool Used: Astah UML

Figure 4 (Component Specification)



Tool Used: Draw.io