Title

Document for Casino Management System as designed by SWAG. It has minimal functionality as of current and covers use cases for logging in, adding and removing funds, and adding to the Work Schedule.

Team

All team members involved contributed to varying degrees:

Henry Felerski

Andrew Gantenbein

Sully Cain

Chris Reeves

Tanner Hoerter

Revision History and Reviews

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Action | Who Reviewed? | Comments |
| 12/14/17 | Created | Henry Felerski | Created document, did Title, Team, and started my section of the Team Summary Time Log. |
| 12/14/17 | Added | Andrew Gantenbein | Wrote deliverable description. |
| 12/14/17 | Revised | Tanner, Chris | Revised diagrams |
| 12/15/17 | Added | Henry Felerski | Continuing Time Log, added deliverables |
| 12/17/17 | Added | Sully Cain | Time Log, Issues |
| 12/17/17 | Added | Tanner Hoerter | Time Log |

Team Summary Time Log

|  |  |  |
| --- | --- | --- |
| Team Member | Total Time (minutes) | Activities |
| Henry Felerski | 1600 | 9/28 Work on Use Case Diagram  10/3 Work on CRC cards  10/4 Continue work on CRC cards  10/5 Group revision of CRC cards  10/26 Created preliminary class diagram, state charts, and wrote up documentation  11/9 Helped with robustness and Collaboration diagrams.  11/13 Reviewed implementation of form design  12/5 Created activity diagram  12/6 Worked on pseudocode and started implementation of entity classes  12/8 Continued my implementation. Switched over to and redid all entity and control classes in C#  12/10 Helped with Boundary Class implementation  12/11 Finalizing implementation before demo |
| Andrew Gantenbein | 1425 | 9/28 Work on Use Case Diagram  10/3 Work on CRC cards  10/4 Continue work on CRC cards  10/5 Group revision of CRC cards  10/26 Created preliminary class diagram, state charts, and wrote up documentation  11/9 Began designing forms  11/13 Reviewed implementation of form design  12/5 Created activity diagram  12/6 Started implementation of form controls  12/8 Continued my implementation. Switched to C# and redid form controls  12/10 Continued form implementation and did some boundary class implementation  12/11 Finalizing implementation before demo |
| Sully Cain | 1200 | 9/28 Work on Use Case Diagram  10/3 Work on CRC cards  10/5 Group revision of CRC cards  10/26 Worked on State charts, helped with class diagrams  11/9 Worked on sequence and collaboration diagrams  11/13 Finished sequence diagrams, helped on writeup  12/5 Wrote pseudocode  12/6 Finished pseudocode, helped on writeup  12/8 Worked on GUI  12/10 Worked on forms, bug tested  12/11 Wrote login user classes and user array |
| Chris Reeves |  |  |
| Tanner Hoerter | 925 | 9/12 Project Proposal  9/28 Use Case Diagram  10/3 CRC cards  10/4 CRC cards individually  10/5 CRC cards revision  10/26 Class diagram, state charts, documentation  11/9 Robustness and Collaboration diagrams  11/13 Sequence Diagram and form design review  12/5 Activity Diagram  12/7 Revise Class Diagram  12/10 Presentation  12/11 Implementation of last Boundary Classes |

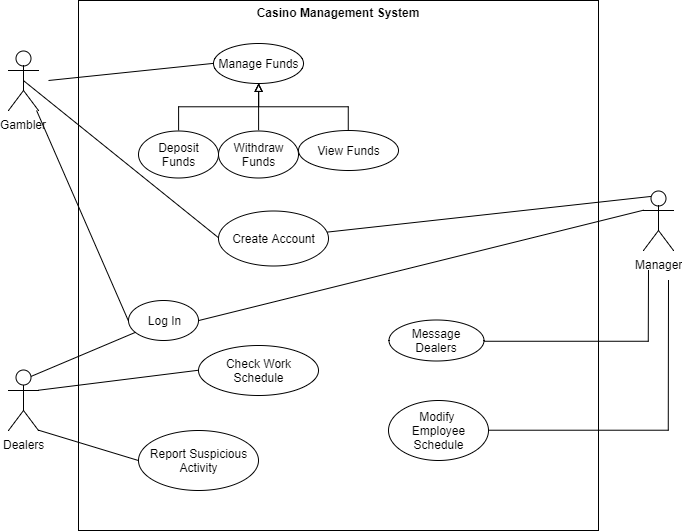
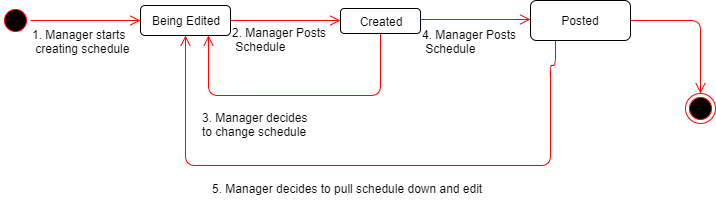
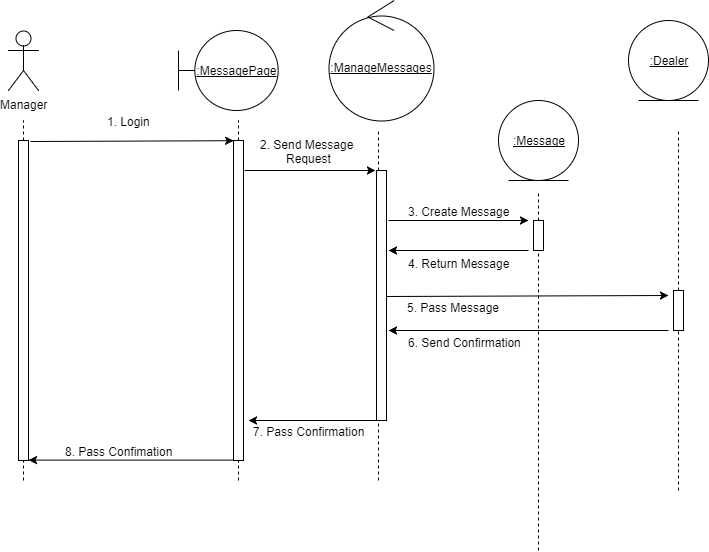
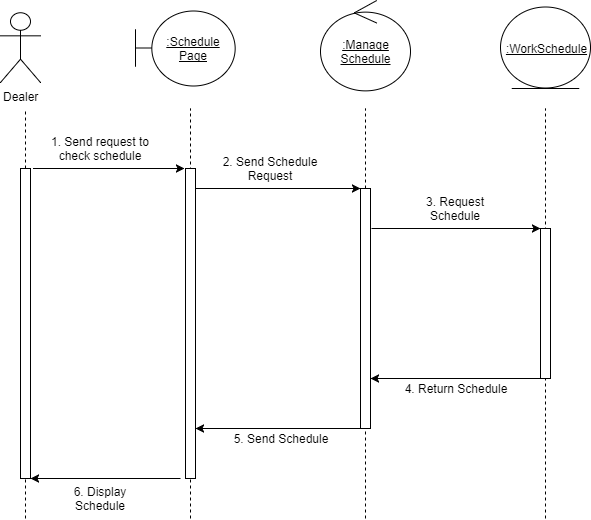
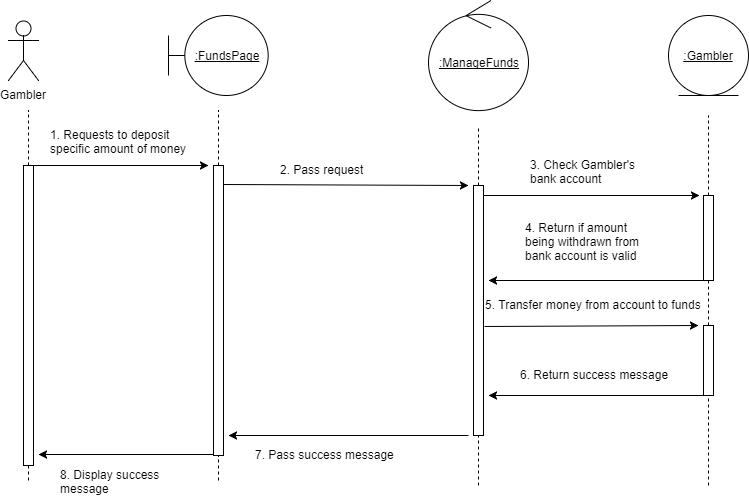
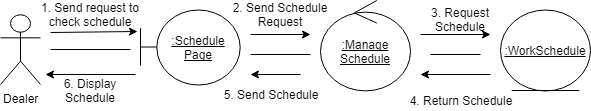
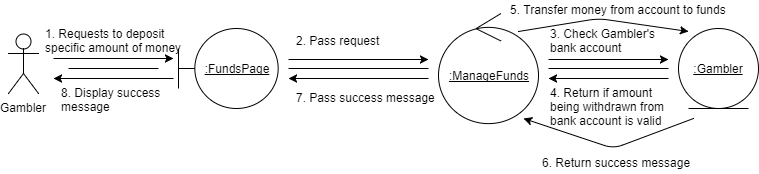
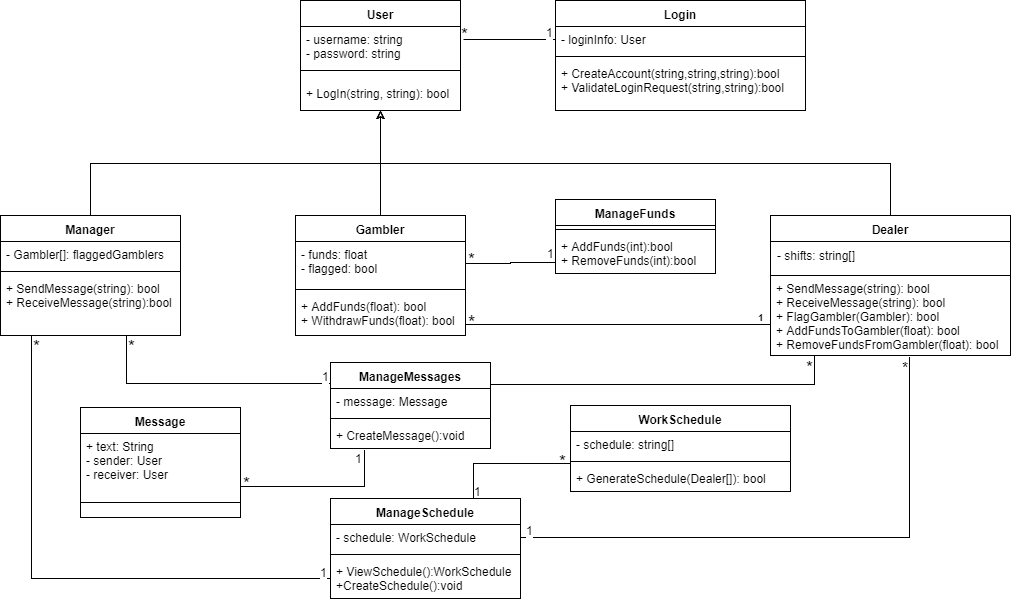
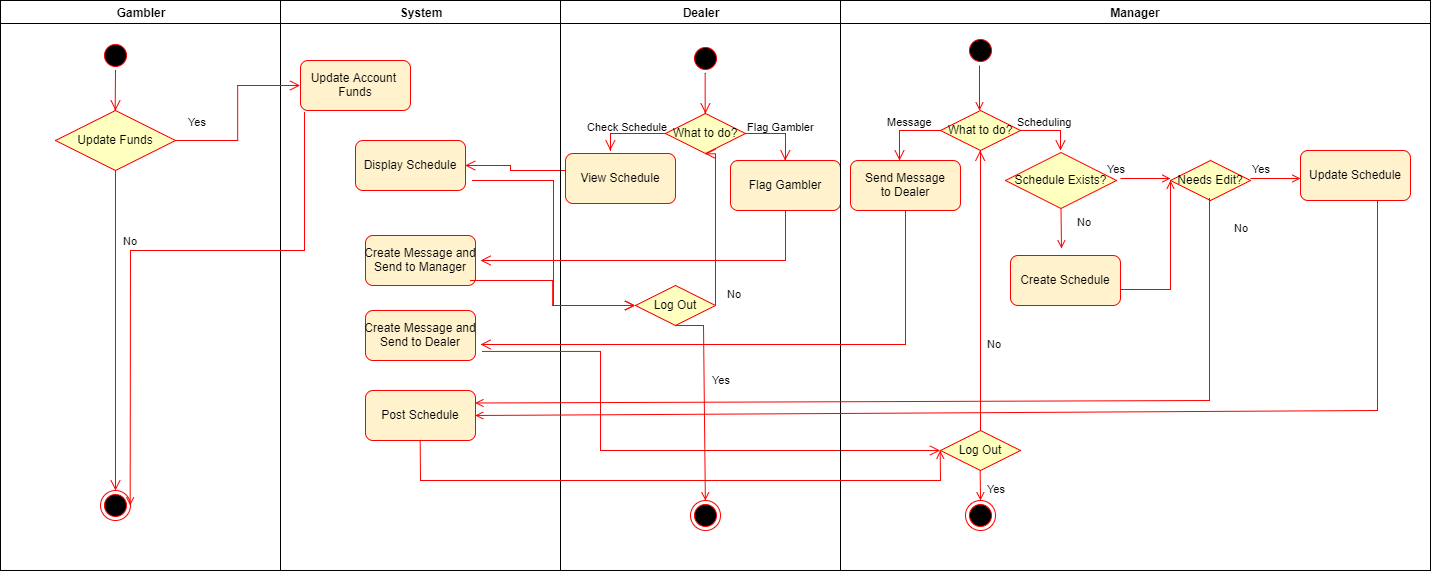
Why are we doing this Deliverable?

We as a group feel like the point of this deliverable is a reflection of the progress that we made throughout the entire semester. Our original idea with the choice of language during our implementation phase was to make the platform comfortable with all of the group members. However, we never really put too much thought into the choice and it proved that we should have put more thought into it when we made the choice. When we went to implement a good majority of our system, we ran into a large amount of issues that revolved around the choice of language because the syntax of C++ was something that we as a group were very unfamiliar with. Also, having split header and cpp files made it harder to manage. So, we made the decision as a group to switch to C# because it was something that was a lot easier to read and the libraries built into the language was more fit to handle the operations that we needed to do in order to accomplish our goal as a group. Also, since we all had experience developing in Visual Studio environment, it was a simple choice to make that the IDE that we wanted to develop our project in.

The prototype that we created for the demonstration in class was not originally the prototype that we had wanted to develop for the demonstration itself. In our minds we wanted to be able to have the functionality of our use case diagram and maybe some other added features along the way. However, since we ran into countless issues during the implementation phase, we were not able to deliver the prototype that we wanted to. The main reason that the prototype that we presented was not up to the standard that we wanted it to be is that a lot of our group members were unavailable to contribute during the implementation phase due to work commitments, family matters, and other school functions. The direct result of this was the rushed implementation phase that saw insufficient amounts of testing, resulting in bugs in a prototype that we presented.

At this moment in time, our group as a whole does not have the desire to continue with the development of this project. However, some of us may, winter break, continue the development of the product and expanding to have different features, such as a database hosted at a remote location or maybe a mobile application for the gamblers using our system.

Deliverables



Issues

During our time working on this project we had many issues with our repository. On several accounts we were met with conflicts and missing files that resulted in large losses of time due to either a misuse or a lack of coordination. We also had difficulty working with C++, our original intended language. Enough issues occurred that made us feel it was worth it to switch over to C# at the last minute in order to save time. Scheduling was another issue in that we had difficulty meeting up to work together throughout the semester. During the implementation period these scheduling issues resulted in redundant and unorganized code that had to be fixed.