**1. How long did the entire project (assignment 2) take you?**  
It took me 5 days to complete the project, I would have liked to have given myself more time but poor time management skills in other subjects left me only being able to start when I did.  
  
**2. What are you most satisfied with?**  
I liked that I was able to keep all GUI aspects in the .kv file. So the application (main) logic was not mixed with styles and layouts.  
  
**3. What are you least satisfied with?**  
Sometimes Kivy for an unknown reason(to me) restores a just pressed item button to 'normal' state after it has been set to 'down' (selected) and on\_release event has been handled. Although it has no effect on the actual internal 'selected' state, but visually it may seem that there is less selected items than is actually the case.  
  
**4. Describe your development process.**  
I tried to have a working application at the end of each step (before each commit). After the first step it was a minimal application that was only able to load CSV file on start and to save it on exit using imported functions from the previous assignment. In the second step I added classes to store items (Item and ItemList). In the third step I turned it to a Kivy application without an interface. In the fourth step I added main window layout and implemented switching between three modes (listing, hiring and returning). In the fifth step I added displaying of items in different colours depending on their availability and implemented selecting of items. In the sixth step, hiring and returning of items were added. And at the end I added popup to add new items.  
  
**5. What worked well in your process?**  
At the beginning of each step I tried to imagine what I wanted to have completed at the end of it. It proved to be quite helpful, as I got a more correct and well-structured result. Using pseudo-code also was helpful in my process.

**6. What about your process could be improved the next time you do a project like this?**  
I would split my code into more, smaller commits.  
  
**7. Describe the main challenges or obstacles you faced and how you overcame them.**  
Finding a way to implement item colouring depending on hired state inside .kv and not to mix it with the application logic. Trial and error brought me to, I believe, a relatively good solution to define a separate button component that calculates its background\_colour depending on hired property directly in the .kv file using colours that are also defined there. Also Kivy uses an interesting way to define background colour for buttons: the background\_colour is not a colour, but a tint to its default background. So it took a time to get it right and find suitable colour combinations.

**8. Describe what resources you used and how you used them to help you complete this.**  
This example app was quite helpful: <https://github.com/CP1404/KivyDemos/blob/master/popup_demo.py>

Also I used the official Kivy documentation for button properties (and for properties in general): <https://kivy.org/docs/api-kivy.uix.behaviors.button.html>

This question on stackoverflow helped to understand how background\_colour property works: <http://stackoverflow.com/questions/20181250/changing-the-background-color-of-a-button-in-kivy>

This article helped with how to implement colouring inside the .kv file: <https://github.com/kivy/kivy/wiki/Styling-a-Spinner-and-SpinnerOption-in-KV>  
  
**9. What were the main things that you personally learned from doing this project?**  
I gained valuable experience in writing a program that has logic/functionality separated from the appearance, I also received practical experience of work with the Kivy framework.