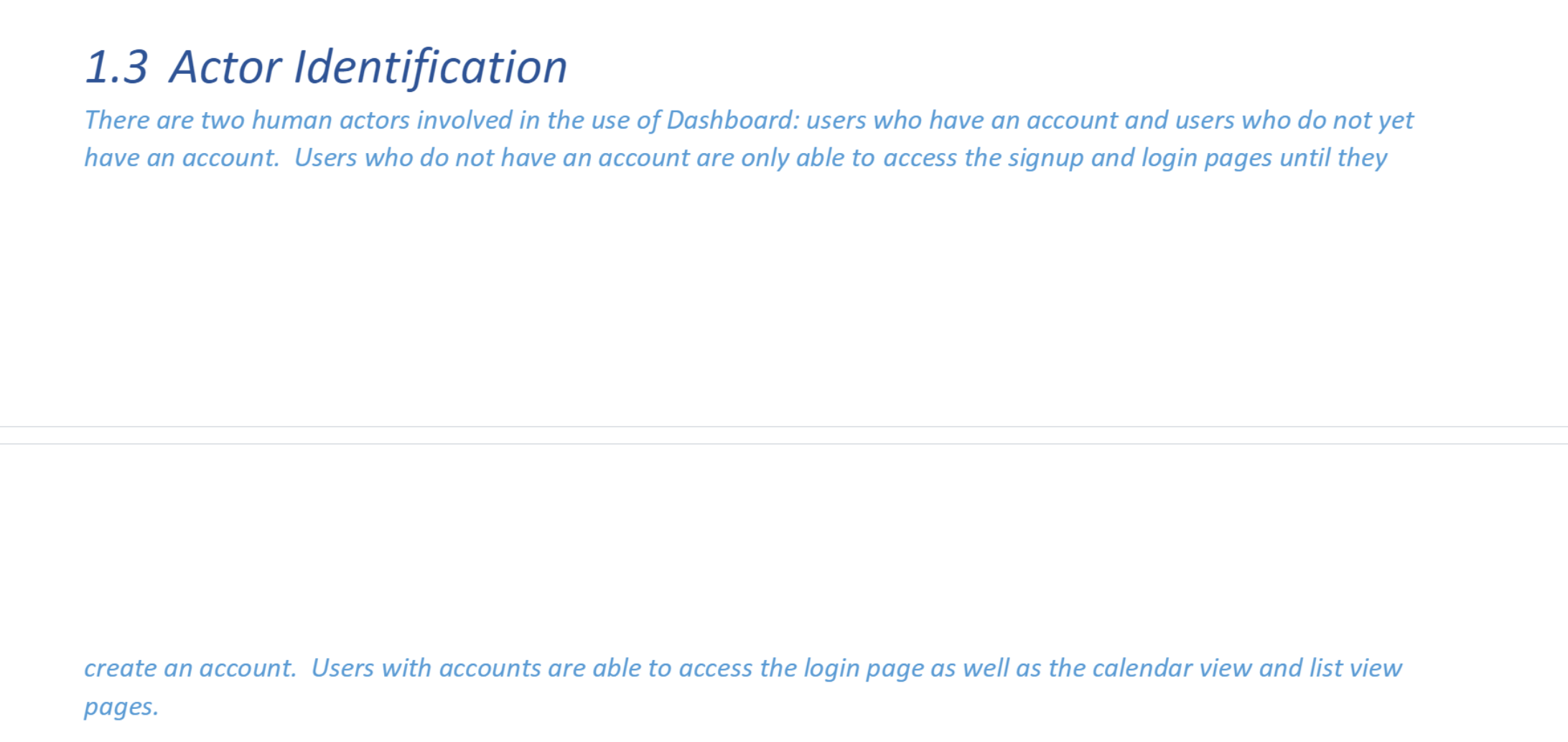
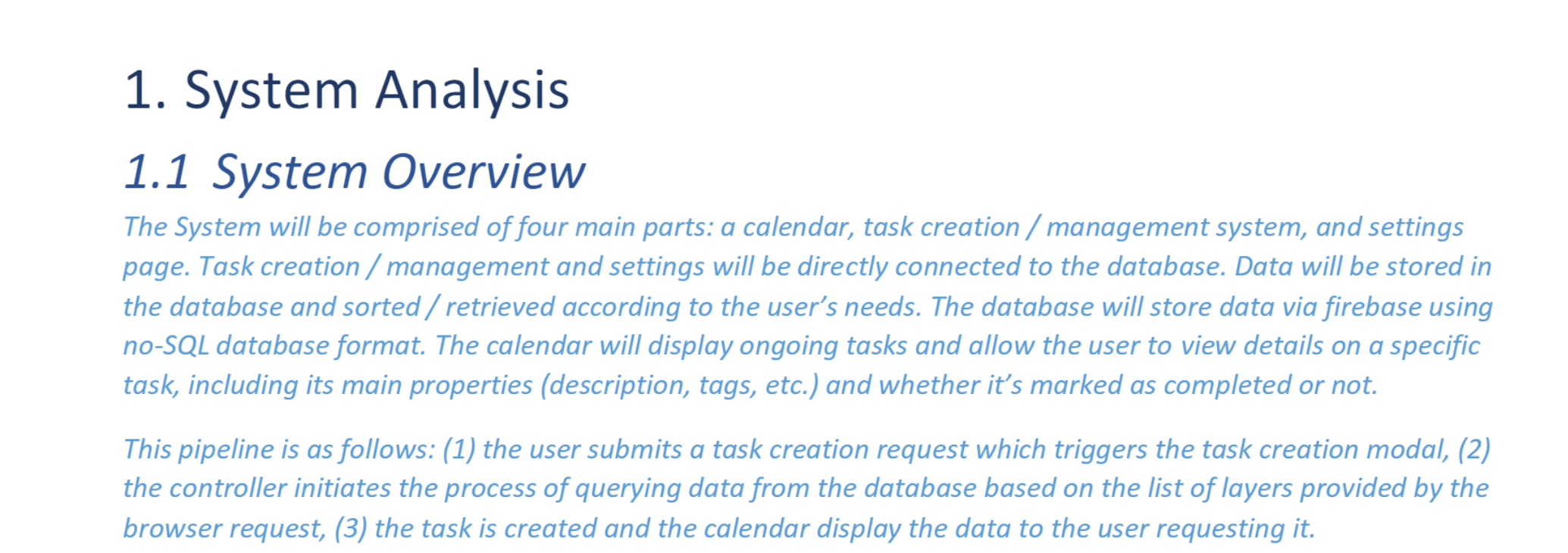
1. Finish functional requirement tables\*\*
2. Add non-functional requirements (ex. Search should return results within 2 seconds, these features are not visible to user) possibly consider caching \*\*
3. Use case diagram\*\*
4. UML diagram
5. Actor list (see lecture 6 slides pg 18)
6. GUI prototypes
7. Screen Mockups

* **~~Does the priority of requirements make sense?~~** ~~yes~~
* **~~Does there appear to be an understanding between functional and non-functional requirements?~~** ~~yes~~
* **~~Do they have an adequate amount of (non-) functional requirements? yes~~**
* **~~Are the system requirements detailed enough?~~** ~~References (figure number) to use case diagrams missing~~
* **~~Does the description make sense?~~** ~~yes~~
* **~~Is the document well written? yes~~**
* **Is the document well formatted?** Consider converting to PDF before uploading it to GitHub (the format is off on the browser). Tables of tables and figures missing. Remove from document examples that apply to this project (e.g., types of system constraints not included)
* **~~Does each requirement’s flow make sense?~~** ~~yes~~
* **~~Are the system constraints appropriate?~~** ~~yes~~
* **Is the use cases diagram detailed enough?** How does the login extend the sign up? Wouldn't it be the other way round? How do the other actors listed interact with the system?
* **Is the use cases diagram formally correct? Do they have the right components, are they used correctly, etc?** The use cases need to match the FRs. Use cases are named using verbs as they represent actions ALSO ADD **FUNCTIONAL DESIGN SEQUENCE DIAGRAM**
* **~~Is UML being used?~~** ~~yes~~
* **Do the evolutionary requirements make sense?** Evolutionary requirements could be included based on the suggestions provided during the presentation
* **~~Comments~~** ~~A role should be given to the people listed, even if "Developer" or "Team member"~~
* ADD FOLDERS TO GITHUB Presentation , SAS, SRS , README
* ADD MORE FAKE TASKS TO GITHUB TO LOOK GOOD
* EVOLUTIONARY REQUIREMENTS ? ? ? ???

SAS TODO :

1. Break up UML diagram into smaller class diagrams (Structural design part)
2. Reuse previous use case diagram and perhaps make another
3. Schematic of the overall system (refer to jerry’s document)
4. Actor identification for each use case diagram (pic example): 
5. System overview: 
6. At least one of the following :
7. 