SE 216 – SOFTWARE PROJECT MANAGEMENT
SOFTWARE MEASUREMENTS DOCUMENT

PROJECT NAME: MediShare

GROUP NUMBER and MEMBERS: Bartu Erdem, Batuhan Can, Erkan Efe Yeşilçimen, Emin Buğra Aksoy, Samet Tolga Esen, Hüseyin Atacan Akgün

Questions to identify measurements:
1. How reliable is this application? 2. What accomplished during certain period of time? 3. How much has the architecture of this system changed ? 4. How well does the project meet user requirements and expectations? 5. What lessons have been learned from project successes and challenges? 6. How effective was the sprints ? 7. Has everyone done their part?
Identified measurements:
Number of tests Number of errors Number of lines of code Number of classes and methods Number of added requirements Number of Git commits Time spend on project Time spend on sprint retrospective User reviews

SE 216 – SOFTWARE PROJECT MANAGEMENT
SOFTWARE MEASUREMENTS DOCUMENT

Measurement storage and collection:
Number of Tests: Before each increment. Real number data.Entered into the "Number of" table by developers. **Number of Errors:** Before each sprint review. Real number data. Add to the "Number of" table by developers. **Time spent on project:** Before each sprint. Real number data. Total number of hours will be entered to the "Duration table" by project leader with the help of time recording logs. **Lines of code:** Before each increment. Real number data.Entered into the "Number of" table by the project leader. **Number of classes methods:** Before each increment. Real number data.Entered into the "Number of" table by developers. **Number of Added Requirements:** After each sprint review. Real number data. Entered into the "Number of" table by project leader. **Time spent on sprint retrospective:** After each sprint retrospective. Real number data. Entered into the project duration by project leader. **Number of Git Commits:** After each sprint review. Real number data. Add to the "Number of" table by developer. **User reviews:** After each increment. Simple text. Entered into the specified user review sheet by developers.

SE 216 – SOFTWARE PROJECT MANAGEMENT
SOFTWARE MEASUREMENTS DOCUMENT

Measurement Type	Description	Example Measurements
Test Numbers	The number of tests conducted to enhance and increase the reliability of our program will be documented in the project table prior to each sprint review. Based on the test results, we will determine whether the program meets the requirements. We will also compare the number of these tests with the number of newly added classes and methods to achieve a close ratio. If the ratio is lower than this, we will review the tests and add new ones after discussing with stakeholders	100, 25, 50, 48
Error message count	The test errors will be reviewed, and the task distribution will be reassessed using git commits. For instance, if an individual consistently receives errors, it may indicate that the workload is too much for one person, or that someone else should take over their tasks.	0, 1, 20, 18
Lines of code	To evaluate the project's progress, we will review the number of codes before each sprint review. This will provide insight into the amount of work completed and the efficiency of resource utilization.	1000, 2000, 1453, 10191
Git Commits	Git commits will be used to evaluate task distribution and performance of an individual.. They will also be used for identifying who received an error during testing.	47, 39, 26, 16

SE 216 – SOFTWARE PROJECT MANAGEMENT
SOFTWARE MEASUREMENTS DOCUMENT

Number methods	This number will be updated prior to each sprint review. This will help estimate the project's progress and determine the number of tests required.	18, 22, 10, 5
Number of changed requirements	By examining this number, we can determine the effectiveness of our planning and identify areas that require attention in future planning. Additionally, it provides insight into our ability to meet user needs.	0, 4, 5
User reviews	After each sprint, user reviews will be kept in a separate file. The file will contain the user's name and feedback. Based on this feedback, requirements will be added and written in different colors, while recommendations will be written in a separate color. The color of the completed requirements will be changed. This approach will demonstrate the extent to which we meet users' needs and can serve as a reference for future projects.	
Time spend on Project	This data will provide information on individual contributions and sprint planning and adherence. We will use this information to optimise project progress by adjusting sprint duration and task allocation. We will attempt to progress the project optimally by re-planning sprint durations and workload distribution.	30 days, 28 days, 21 days
Time spend on sprint retrospective	By examining the time spent in the sprint retrospective, it will become apparent whether the sprint planning was effective. If the retrospective period is lengthy, it suggests that the sprint planning was flawed from the outset.	2 hours, 1 hour

	This indicates that the developers may not have fully grasped the Scrum framework or that there may be miscommunication between them.	

Sprint	Tests	Errors	Lines of code	Methods	Classes	Changed Requirements	Git Commits
1	80	8	1627	8	1	0	24
2	100	4	2433	19	2	4	37

Sprint	Time spend on project	Time spend on retrospective
1	30 days	2 Hours
2	28 days	1 Hour

SE 216 – SOFTWARE PROJECT MANAGEMENT
SOFTWARE MEASUREMENTS DOCUMENT

User Review Sheet Example:

User A:

This requirement must be added.

This requirement was added by a user and completed during the project.

This part requires attention.

User B:

This requirement must be added.

This requirement was added by a user and completed during the project.

This part requires attention.

User C:

This requirement must be added.

This requirement was added by a user and completed during the project.

This part requires attention.