Functional Requirements

MODULE	Requirement	Workload	Description	Estimated Time [h]	Total[h]
0: Preparations	Preparation	Prepare framework and structure MV	Prepare namespaces, folder structure, symbolic links and cmake file	3	**3**
		Database for commands	Implement a database with all basic OBD commands (database design and implementation) for each OBD tool technology (e.g. ELM327)	10	**20**
		Basic framework	Structure of the software as a basic framework basic classes like representing the trouble code, the connection and the recordings	10	
3: Trouble Code Management	Troube code management	Read trouble codes	Read the trouble codes from the ECU memory	10	**20**
		Delete trouble codes	Clear the trouble code memory in the ECU	10	
4: Data Management	Data management	Read data	Read the sensor data, frame data and testresults	20	**70**
		Get vehicle information	Get the vehicle information from the ECU	10	
		Play/Record sensordata	Record the gotten sensordata when and play a recording	40	
	Sensordata	CSV input	Implement parsing of CSV files, where each row will have a timestamp and each column will represent a sensor	15	**30**
		Play	Play "recording" of sensordata by timestamp	5	
		Observer Pattern	Implement the Sensordata class as observer which classes can subscribe to	10	
5: Communication	Systeminterface + Communication Simulation	Mounting virtual serial port	Mounting a virtual serial port from the simulation emulating the OBD Tool hardware of the ELM327 and being able to use it with OBD softw	30	**75**
		Interface for communication	Implementing interface for communication with serial port in the simulation	15	
		Configuration	Implementing configuration of serial port (baud rate etc..) via XML for OBD Tool and Simulation	30	
6: GUI + Usability	GUI Simulation	Preparation	Set up basic GUI elements from GTK API and implement basic structure	15	**60**
		Troublecode memory	Create GUI control window for troublecode memory to directly alter its contents	10	
		Controller	Create controllers between model and GUI	10	
		Sensordata	Implement window for displaying and configuring sensordata	12	
		OBD hardware	Implement the OBD hardware window, selection of ELM type, communication settings etc..	13	
	GUI OBD Tool	Trouble code window	Make a view for representing the trouble codes	25	**75**
		Data view	Implement a view representing the data read from ECU	25	
		Communication view	Implement a communication view for the configuration of the connection to the ECU and ECU simulation	25	
7: XML and Database Management	Errorcode database + Response	Create database (base structure)	Implement a SQL script creating the database of trouble codes (not all for now)	13	**60**
		SQL interface	Get SQL API library and implement interface for communication with software	12	
		Troublecode design pattern	Implement factory pattern (SQL entry to trouble code object), trouble coude base class and research the response structure of the CU in O	15	
		Troublecode memory class	Implement container class representing the troublecode memory in a CU	10	
		Communication interface	Implement an interface, which is able to get requests (from OBD interface) and send responses (to OBD interface)	10	
	Communication	XML Reader	Implement XML interface for framework	15	**15**
9: Hardware configuration	Hardware Issues	Mount hardware and driver Issues	Mounting ELMs and Connection issues Linux	25	**40**
		Selection of Serial Port	Selection of serial port in OBD Tool	15	
				Total	468

Non Functional Requirements

1	Free of charge
2	Open source
7	Programing style, OOP and no hardcoded values
8	WWH-OBD and EOBD standards