	Title	Description	Assigned to	Workload (hours)	Deadline	Current progress	Remaining
TrainServer:				76		16%	42.30
	Client requests journey information	**Create weighted and directed graph from traffic dataset. And calculate best route from A to B**		**52**		**40%**	**18.80**
		Check that graph exists, else get new graph and wait to respond	James	2		0%	2.00
		Setup graph, evaluate nodes, and score them	James	16		80%	3.20
		Find the shortest path from A to B with the set up graph	James	22		70%	6.60
		Backtrack the shortest path and add those nodes to a Journey Object.	James	10		50%	5.00
		Return the journey object to the user.	James	2		0%	2.00
	Charge users when abscent	**A short while after leaving a station, the train/bus must search for user connections. All abscent "active" users are charged**		**12**		**8%**	**11.50**
		Send out UDP signal	James	1		50%	0.50
		Create an array with all incoming user connections.	James	2		0%	2.00
		Compare the new array with the Active Users array, delete from new array at match	James	3		0%	3.00
		When finished comparing, calculate charge for remaining users in the new array.	James	3		0%	3.00
		Send charge to Server.	James	1		0%	1.00
		Remove users in new array from Active Users Array.	James	2		0%	2.00
	Calculate charge	**calculating what to charge a user when that user leaves the train/bus**		**12**		**0%**	**12.00**
		Finding the dearest zone a passenger has been to.	James	4		0%	4.00
		finding the start zone	James	4		0%	4.00
		Calculating the cost per zone times the number of zones between start and destination	James	4		0%	4.00
Client:				23		22%	17.50
	Logon to trainserver	**the client will listen for a UDP signal, and read the TCP address to connect to.**		**6**		**43%**	**2.60**
		The client recieves a UDP signal from the train server.	James	3		80%	0.60
		The client translates the signal to a TCP address and sends a connect command with his UserID	James	2		50%	1.00
		When a response comes the client will read the received user information and display it	James	1		0%	1.00
	get fastest journey to destination	**The client will request the shortest path from the trains current location to the required destination.**		**8**		**0%**	**8.00**
		The client sends a request for the route.		2		0%	2.00
		When the response is gotten (a Journey object), the response is read and the nodes (stops) are shown with time to destination and node names.		6		0%	6.00
	UI	**An suggestion for an interface on the client "PDA" showing status to a conductor**		**9**		**23%**	**6.90**
		Designing the layout	James	3		70%	0.90
		Listening for logged in status	James	2		0%	2.00
		Get journey screen with destination field and button.	James	4		0%	4.00
Server:				68		6%	65.80

Make a graph from the traffic database	**the main server will make a graph from the traffic database whenever a change is made to it.**		**23**	**0%**	**23.00**
	A change to the traffic database will trigger and reset a slightly delayed timer.		3	0%	3.00
	the timer will, when elapsed request and update of the traffic network graph.		2	0%	2.00
	The traffic database is read and each stop is assigned a number of routes corresponding to the relation db.		16	0%	16.00
	The new graph is distributed around to train and bus servers		2	0%	2.00
Charge a user	**Trainservers will request a user to be charged in the user database**				
	Scenario A: RMI - Advantage: shows RMI in use		**21**	**0%**	**21.00**
	An rmi call from the train server requests a user object to charge. The server searches for an cached user object in a user array	Lukas	15	0%	15.00
	if the object exists, the RMI handler will charge that user object	Lukas	2	0%	2.00
	if it doesn't exist the object is made from the database and put into the array. Maybe use a map to guarantee uniqueness.	Lukas	4	0%	4.00
	Scenario B: Producer/Consumer - Advantage: is simple and stabile		**6**	**0%**	**6.00**
	A call to charge a user is stored in a concurrent array	Lukas	2	0%	2.00
	A single looping thread reads the array and updates the database with the charges.	Lukas	4	0%	4.00
Manage databases	**Creating, deleting, updating, and getting entries in database tables**		**18**	**23%**	**15.80**
	A single entry to each database that handles connection and takes in query strings	James	1	100%	0.00
	A function for addingUser	Sudhir	3	0%	3.00
	A function for deletingUser	Sudhir	2	0%	2.00
	A function for chargingUser	Lukas	4	0%	4.00
	a function for UpdatingUser	Sudhir	4	0%	4.00
	a function for getting user	Sudhir	2	0%	2.00
	a function for getting graph	James	2	60%	0.80