

## Message Analysis Table – Systems Analysis Project

Table 1 – To register for a membership

Sequence Number	Caller	Callee	Msg Name	Msg Type	Msg Param	Msg Constraint
1	Member	Registration UI	EnterDetails()	Synchronous	userDetails	The member is not registered
2	Registration UI	MemberController	CreateMember()	Synchronous	userDetails	The inputs must be valid
3	MemberController	Membership	NewMembership()	Synchronous	membership Data	Must not have a membership
4	MemberController	Payment	ProcessPayment()	Synchronous	amount	Payment is required
5	Payment	MemberController	PaymentResult()	Return	status	Payment has to succeed
6	MemberController	Database	SaveMember()	Synchronous	memberDetails	Must pay before it gets saved
7	Database	MemberController	Acknowledge()	Return	N/A	Saved successfully
8	MemberController	Registration UI	ShowConfirmation()	Return	N/A	The registration is complete

Table 2 – To book a spot in a class

Sequence Number	Caller	Callee	Msg Name	Msg Type	Msg Param	Msg Constrain
1	Member	Booking UI	selectClass()	Synchronous	classID	Must already be logged in
2	Booking UI	Booking Controller	bookingRequest()	Synchronous	ClassID, memberID	Class has to exist
3	Booking Controller	Gym Class	checkAvailability()	Synchronous	classID	Must check class capacity
4	Gym Class	Booking Controller	returnAvailability()	Return	availability	Must be available
5	Booking Controller	Booking	newBooking()	Synchronous	MemberID classID	Must be Available
6	Booking Controller	Database	saveBooking()	Synchronous	bookingDetails	Booking must have been created
7	Database	Booking Controller	Acknowledge()	Return	N/A	Booking saved successfully
8	Booking Controller	Booking UI	bookingConfirmation()	Return	N/A	The booking is complete

Table 3 – Renew a membership

Sequence Number	Caller	Callee	Msg Name	Msg Type	Msg Param	Msg Constraint
1	Member	MembershipUI	clickRenew()	Synchronous	N/A	The member is logged in
2	MembershipUI	MembershipController	renewRequest()	Synchronous	memberID	The membership must exist
3	MembershipController	Payment	processPayment()	Synchronous	amount	Valid payment method
4	Payment	MembershipController	paymentStatus()	Return	status	Must succeed
5	MembershipController	Membership	renew()	Synchronous	N/A	Payment has to have happened
6	Membership	MembershipController	updateExpiry()	Return	newExpiry	The expiry is updated
7	MembershipController	Notification	sendNotification()	Synchronous	memberEmail	Only after renewal

8	Notification	MembershipController	N/A	Return	N/A	The notification is sent
9	MembershipController	MembershipUI	ShowConfirmation()	Return	N/A	UI shows result

Table 4 – To create a new gym class

Sequence Number	Caller	Callee	Msg Name	Msg Type	Msg Param	Msg Constrain
1	Trainer	TrainerUI	EnterDetails()	Synchronous	classData	The trainer is logged in
2	TrainerUI	ClassController	submitClass()	Synchronous	classData	The details must be valid
3	ClassController	GymClass	newGymClass()	Synchronous	classData	Must not already exist
4	ClassController	Database	saveClass()	Synchronous	classObject	Use correct object
5	Database	ClassController	acknowledge()	Return	none	Save succeeds

6	ClassController	TrainerUI	showConfirmation()	return	none	UI shows confirmation
---	-----------------	-----------	--------------------	--------	------	-----------------------

Table 5 – Admin generates a report

Sequence Number	Caller	Callee	Msg Name	Msg Type	Msg Param	Msg Constraint
1	Administrator	AdminUI	requestReport()	Synchronous	reportType	The Admin is logged in
2	AdminUI	ReportController	generate()	Synchronous	reportType	The type is valid
3	ReportController	Database	getData()	Synchronous	theParameters	Data must exist
4	Database	ReportController	response(data)	Return	data	It succeeds
5	ReportController	Report	generateReport(data)	Synchronous	data	Data gets retrieved
6	Report	ReportController	sendFile()	Return	FilePath	File gets

						generat ed
7	Database	AdminUI	showFile()	Return	FilePath	Ui display s file