Help to complete the tasks of this exercise can be found on the chapters from ch. 13 “Setting Up a MongoDB Database”, ch. 14 “Building Models with Mongoose”, and ch. 15 “Connecting Controllers and Models” of our course book “Get Programming with Node.js” by Jonathan Wexler. The aims of the exercise are to learn working with models and to learn using a database as a permanent data storage of your web application. You will also learn to utilize promises to help you with the asynchronous programming.

This time you can earn some extra points. Due to having a three chapter’s reading area and more than a usual number of tasks the maximum number of points available is 13.

Embed your theory answers, drawings, codes, and screenshots directly into this document. Always immediately after the relevant question. Return the document in ItsLearning return box by the deadline.

It’s also recommendable to use Internet sources to supplement the information provided by the course book.

1. Install and setup MongoDB. Connect your recipe application to MongoDB database management system and to recipe\_db database. (1 point):
   1. Install and test MongoDB database management system on your Ubuntu virtual machine. Create also the directory /data/db. Please, note that if you set authentication you’ll need to make changes to some of the (You can also install it on another platform.) (0,5 points)
   2. Install mongodb package to your Node.js recipe application?
   3. Follow the instruction give in the listing 13.5 in your course book and create the connection to the recipe\_db ? Do not use the name main.js, use just mongo-test.js.
   4. Add a couple of contacts to your contacts collection according to the instructions given in the listing 13.6. Make certain that you add your own (fake) contact information too. (0,5 points)
2. What does it mean that Mongoose is an ODM? (1 point)
3. Create an application that can be used to create subscribers by utilizing Mongoose. You can look for help from the listings 14.1 to 14.4. Make certain that you add one more feature to the scheme. The feature is called vip and it is of type Boolean. (2 points)

1. Modify your recipe application by moving the subscribers schema and the subscribers model into a separate file in the models folder. Search help from the listings 14-5 and 14.6. Make certain that you query your own contact information. (2 points)
2. Follow the instructions given in the listings 14.1 to 14.6, but this time create a schema and a model to the recipe item. (1 point)
3. Follow the instructions given in the listings 15.1 to 15.3 and use models, views and controllers (MVC) together to display the subscribers on a web page. Please make certain that also the vip status is displayed. (2 points)
4. Follow the instructions given in the listings 15.4 to 15.6 and make certain that the user can add new subscribers from the contact page. Please make certain that also the vip status can be saved. (2 points)
5. Follow the instruction given in the listings 15.7 and 15.8 and change your subscribers controller to be promise-based. (2 points)