METU CENG491 2015 FALL

START-UP DOCUMENT

<G31P36>

Group Name: Malum

Project Name: *Guideal*

1. System Architecture

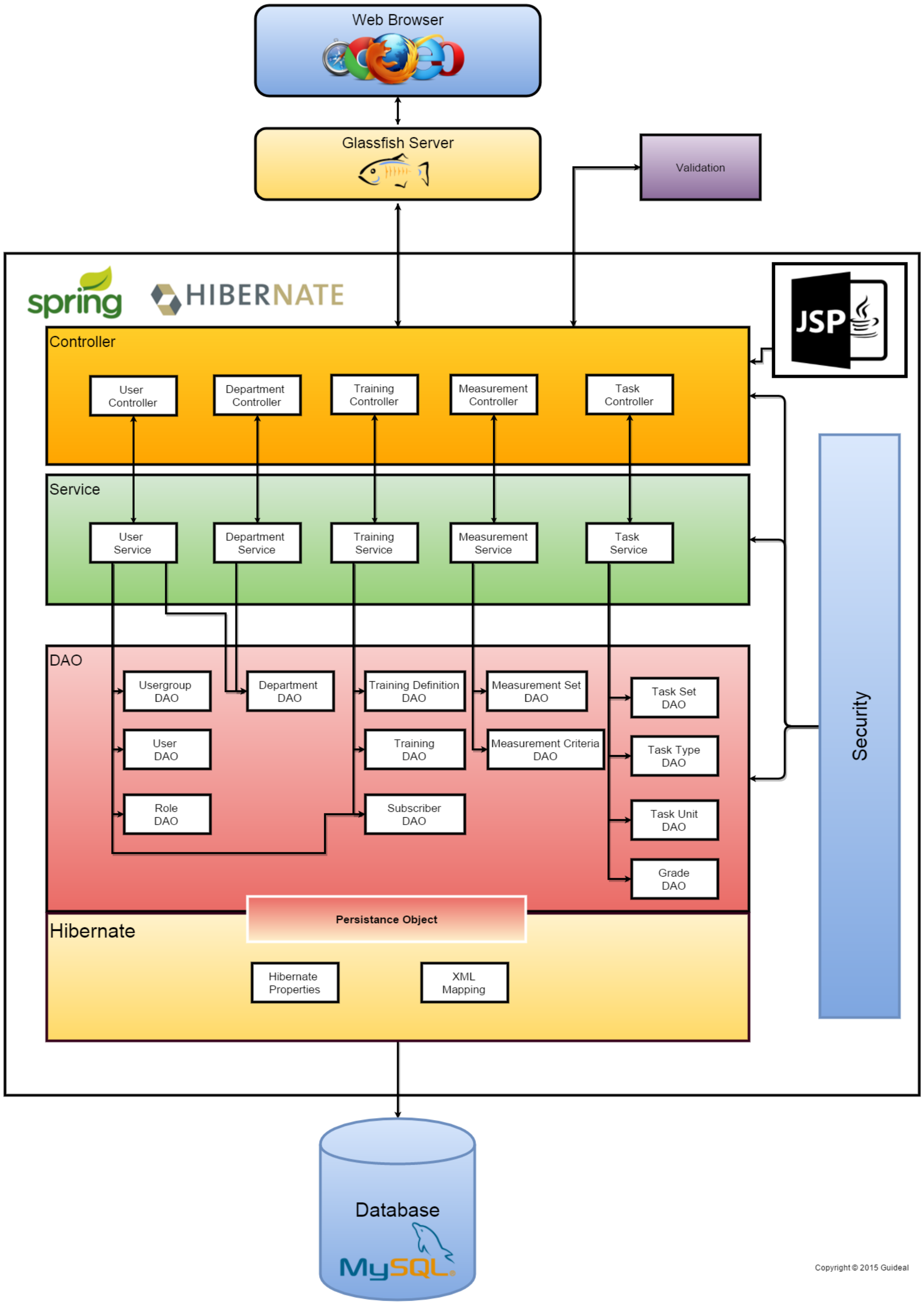

Figure 1 : System Architecture

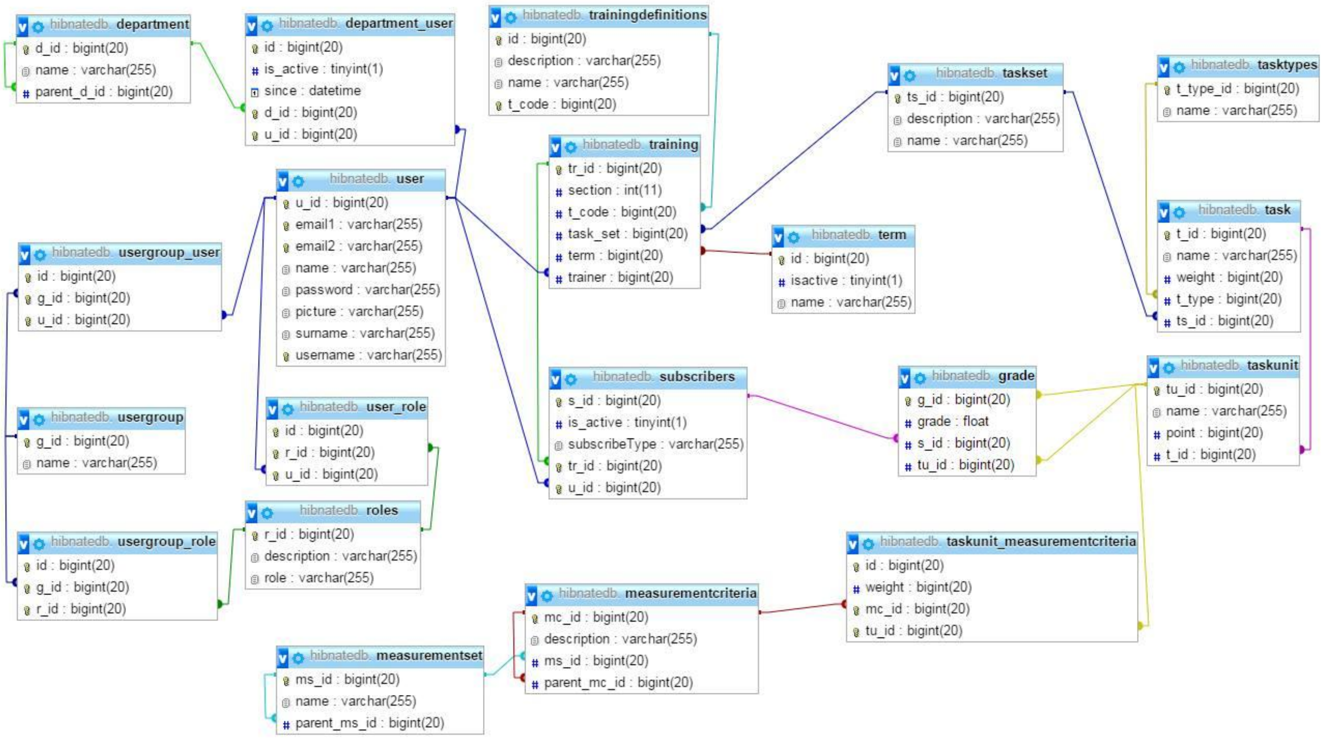

Figure 2: Entity Relation Diagram

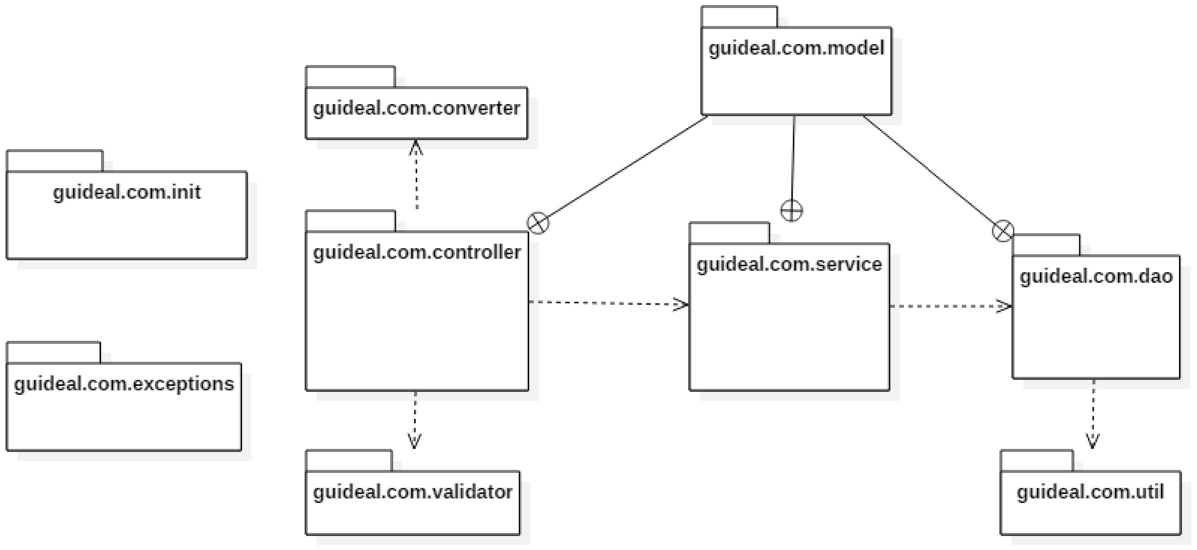

Figure 3: Package Diagram

Users will access our system by using web browsers. Our system will run on all modern web browsers. Users will send their requests via Glassfish Server and Java Server Pages(JSP).

When users enter our system, our java controller classes automatically handle requests by using services provided in model side and by sending results back to the view side. As you can see in ***Figure1***, for each logical system we have one releated controller class. Therefore, each different logical request will be handled by different controller.

Our controller classes are responsible for URL-mapping, user authorization, form validation and calling related functions from service layer. After retrieving data from service classes, it sends the data to JSP and our JSP (view side) represents data to the user.

When it comes to service layer, we have 5 services just like in controller layer. Service layer is responsible for representing data which comes from DAO layer. Service layer provides related functions from DAOs and prepares the final data. Each service class and its related DAOs are shown in ***Figure 1***. Also, service layer collaborates with controller layer when security in the application is considered.

Finally, our database and web base application is connected to database with DAO layer. Each DAO layer class is responsible for spesific table. DAO classes add, delete, update, get entites from database and execute many variable queries for variable needs.

Final picture is that, user sends requests from view layer (JSPs). Our controllers take them and validate them. After validation, releated functions from service layer are called by controllers. Service layer collects data from DAO layer. Meanwhile, DAO layer accesses database and gives desired data to service layer. When service layer represents data, it gives data to controller layer back. After that, controller layer meets user responds user request and waits for another user request.

In **Figure3**, we try to show our organization. In model package classes will represent our database entities (**Figure 2**). Other package names are self-explanatory and we already discuss about them.

2. Tentative Time Plan

- *Identify and itemize all tasks to be performed as a team in the first semester. Assign a unique TaskID for each task. Give a short name and brief description for each identified task.*

TaskID	Short Name	Description
SYS_INIT	System initialize	System environment will be setup in this task. Hibernate, Spring and Glassfish Server will be initialized.
Mark_Res	Market research	Market research will be done in this step. Similar projects and system will be analyzed in this step.
U_M	User_Model	User, Role, UserRole, Usergroup, UserUsergroup, UsergroupRole classes will be implemented according to Figure 2.
U_D	User_DAO	UserDAO, RoleDAO, UserRoleDAO, UsergroupDAO, UserUsergroupDAO, UsergroupRoleDAO classes will be implemented. Their aim is that they will access the database to add, remove, update, get related table entities.
U_S	User_Service	UserService class will be implemented. Its aim is that, it will call related DAO classes to serve desired information.
U_C	User_Controller	UserController class will be implemented. Its aim is that, it will organize user pages. It will take requests, validate them and check authentication. After that, calls related UserService functions to get desired data.
U_V	User_View	User pages will be created and related JSP files will be implemented.
D_M	Department_Model	Department, DepartmentUser classes will be implemented according to Figure 2.
D_D	Department_DAO	DepartmentDAO, DepartmentUserDAO classes will be implemented. Their aim is that they will access the database to add, remove, update, get related table entities.
D_S	Department_Service	DepartmentService class will be implemented. Its aim is that, it will call related DAO classes to serve desired information.
D_C	Department_Controller	DepartmentController class will be implemented. Its aim is that, it will organize department pages. It will take requests, validate them and check authentication. After that, calls related

		DepartmentService functions to get desired data.
D_V	Department_View	Department pages will be created and related JSP files will be implemented.
TR_M	Training_Model	Training, TrainingDefinition, Subscriber, Term classes will be implemented according to Figure 2.
TR_D	Training _DAO	TrainingDAO, TrainingDefinitionDAO, SubscriberDAO,TermDAO classes will be implemented. Their aim is that they will access the database to add, remove, update, get related table entities.
TR_S	Training _Service	TrainingService class will be implemented. Its aim is that, it will call related DAO classes to serve desired information.
TR_C	Training _Controller	TrainingController class will be implemented. Its aim is that, it will organize training pages. It will take requests, validate them and check authentication. After that, calls related TrainingService functions to get desired data.
TR_V	Training _View	Training pages will be created and related JSP files will be implemented.
TA_M	Task_Model	Task, TaskSet, TaskType, TaskUnit, Grade classes will be implemented according to Figure 2.
TA_D	Task _DAO	TaskDAO, TaskSetDAO, TaskTypeDAO, TaskUnitDAO, GradeDAO classes will be implemented. Their aim is that they will access the database to add, remove, update, get related table entities.
TA_S	Task _Service	TaskService class will be implemented. Its aim is that, it will call related DAO classes to serve desired information.
TA_C	Task _Controller	TaskController class will be implemented. Its aim is that, it will organize task pages. It will take requests, validate them and check authentication. After that, calls related TaskService functions to get desired data.
TA_V	Task _View	Task pages will be created and related JSP files will be implemented.
M_M	Measurement_Model	MeasurementSet, MeasurementCriteria, TaskUnitMeasurementCriteria classes will be implemented according to Figure 2.
M_D	Measurement _DAO	MeasurementSetDAO, MeasurementCriteriaDAO, TaskUnitMeasurementCriteriaDAO classes will be

		implemented. Their aim is that they will access the database to add, remove, update, get related table entities.
M_S	Measurement _Service	MeasurementService class will be implemented. Its aim is that, it will call related DAO classes to serve desired information.
M_C	Measurement_Controller	MeasurementController class will be implemented. Its aim is that, it will organize measurement pages. It will take requests, validate them and check authentication. After that, calls related MeasurementService functions to get desired data.
M_V	Measurement _View	Measurement pages will be created and related JSP files will be implemented.

- *Construct your time plan as a simplified Gantt chart, as shown in the following table.*

	Iteration1	**Iteration2**	**Iteration3**
SYS_INIT	■	■	■
Mark_Res	■	■	■
U_M	■	■	■
U_D	■	■	■
U_S	■	■	■
U_C	■	■	■
U_V	■	■	■
D_M	■	■	■
D_D	■	■	■
D_S	■	■	■
D_C	■	■	■
D_V	■	■	■
TR_M		■	■

TR_D		■	■
TR_S		■	■
TR_C		■	■
TR_V		■	■
TA_M		■	■
TA_D		■	■
TA_S		■	■
TA_C		■	■
TA_V		■	■
M_M			■
M_D			■
M_S			■
M_C			■
M_V			■

3. Deliverables

- *Identify and list all deliverables of your project for the first 3 sprints.*
- *A deliverable is some component or sub-component, which is running and demonstrable to your assistant and your supervisor. That deliverable is of course subject to improvement over time.*
- *Fill in the following table:*

Deliverable	**Description**	**When? (Sprint#)**
D1	Market research and system initialization will be done in this deliverable. We can inform people about marketing of this project and system components such Hibernate, Spring and Glassfish	Sprint 1
D2	User pages will be done in this step. In this deliverable, we can show adding new user, updating them and deleting them.	Sprint1
D3	Department pages will be done in this step. In this deliverable, we can show user pages and department pages. Moreover we can show add, remove, update and get operations.	Sprint1
D4	Traning pages will be done in this step. In this deliverable, we can show training, department, user pages. And their operations	Sprint2
D5	Task pages will be done in this step In this deliverable, we can show task, training, department, user pages. And their operations.	Sprint2
D6	Measurement pages will be done in this step. In this step, we can show full system. This system will have all database access. User, Department, Training, Task, Measurement pages will be done. After that stage, user authentications will be done. We can show simply demo in that stage. In this demo, users can logging and they can surf on their pages. Courses can be added, and users can register these courses. After that stage, our background system will be done. We will be ready for our project intelligent part which are reporting systems, recommendation system and implementing new plugins.	Sprint3

4. Workload Distribution

Fill in the following table to distribute the workload for the first semester among your team members.

	Sprint - I	**Sprint - II**	**Sprint - III**
SemihAktaş	U_M, U_D, D_D, D1, D2, D3	TR_M, TR_D, TA_M, D4, D5	M_D, D6
Yusuf MücahitÇetinkaya	U_S, D_M, D_S, D1, D2, D3	TR_S, TA_D, TA_S, D4, D5	M_M, M_S, D6
GörkemÖzer	U_C, D_C, D1, D2, D3	TR_C, TA_C, D4, D5	M_C, D6
EmreKülah	U_V, D_V, D1, D2, D3	TR_V, TA_V, D4, D5	M_V, D6

Simply we are using MVC pattern.

Model	**->SemihAktaş, Yusuf MücahitÇetinkaya**
View	**->EmreKülah**
Controller	**->GörkemÖzer**