ALMU-järjestelmä

Suunnitteludokumentti

Helsingin yliopisto
TKTL
Ohjelmistotuotantoprojekti
Projektiryhmä:
Peter Ahlberg, Marika Korhonen, Tomi Kuittinen, Iikka Meriläinen, Jukka Narkiniemi

SISÄLLYS

1 Johdanto

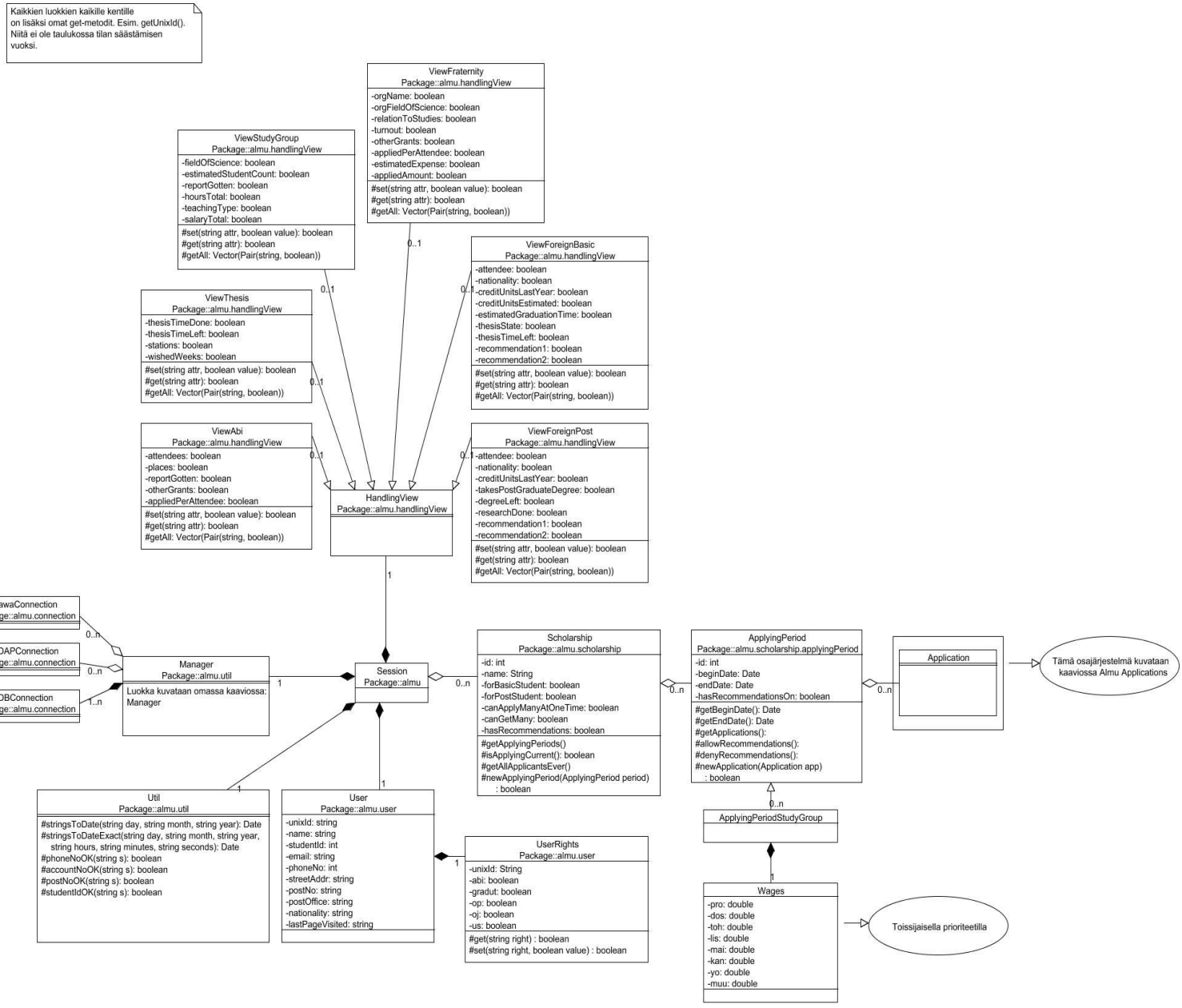
Suunniteludokumentti kuvaa toteutettavan järjestelmän sellaisella tasolla, että toteutus on suoraviivainen. Suunnitteludokumentin perusteella jokaisen ryhmän jäsenen täytyy voida keskittyä oman osuutensa koodaamiseen, ilman että tarvitsee jatkuvasti päivitellä osuuksien välisiä rajapintoja.

Ensin kuvataan koko järjestelmäarkkitehtuuri, minkä jälkeen järjestelmän osajärjestelmät kuvataan yksi kerrallaan. Kunkin osajärjestelmän kohdalla kuvataan luokat, joiden oliot toteuttavat osajärjestelmän.

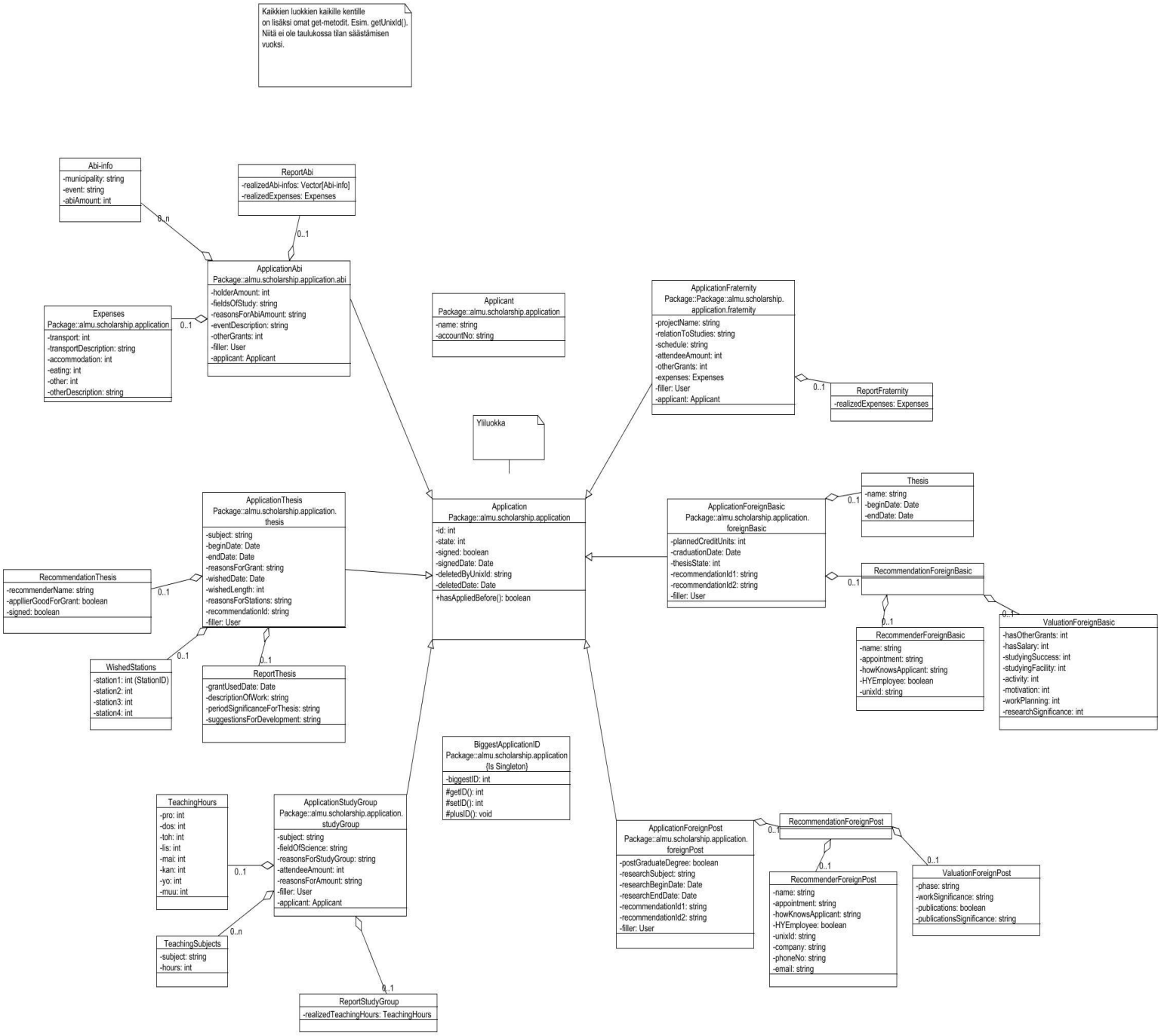
2 Sanasto

Sana	*Suomennos / selitys*
Thesis	Pro gradu
Study group	Opintopiiri
Fraternity	Opiskelijajärjestö
Application foreign	Ulkomaalaisstipendin apurahahakemus
Basic student	Perustutkinto-opiskelija
Post graduate student	Jatkotutkinto-opiskelija
Handling view	Käsittelynäkymä
Scholarship	Apuraha / stipendi
Applying period	Apurahan hakukerta
Turnout	Osallistujamäärä
Credit unit	Opintopiste
Degree	Tutkinto
Realized	Toteutuneet
Reasons for	Perustelut jollekin
Craduation	Valmistuminen
Appointment	Virka / työnimike
HYEmployee	Helsingin yliopiston työntekijä
Summary	yhteenveto

3 Järjestelmäarkkitehtuuri



4 Osajärjestelmä: Almu applications



4.1Viestinvälitysrajapinta

-Application-luokka

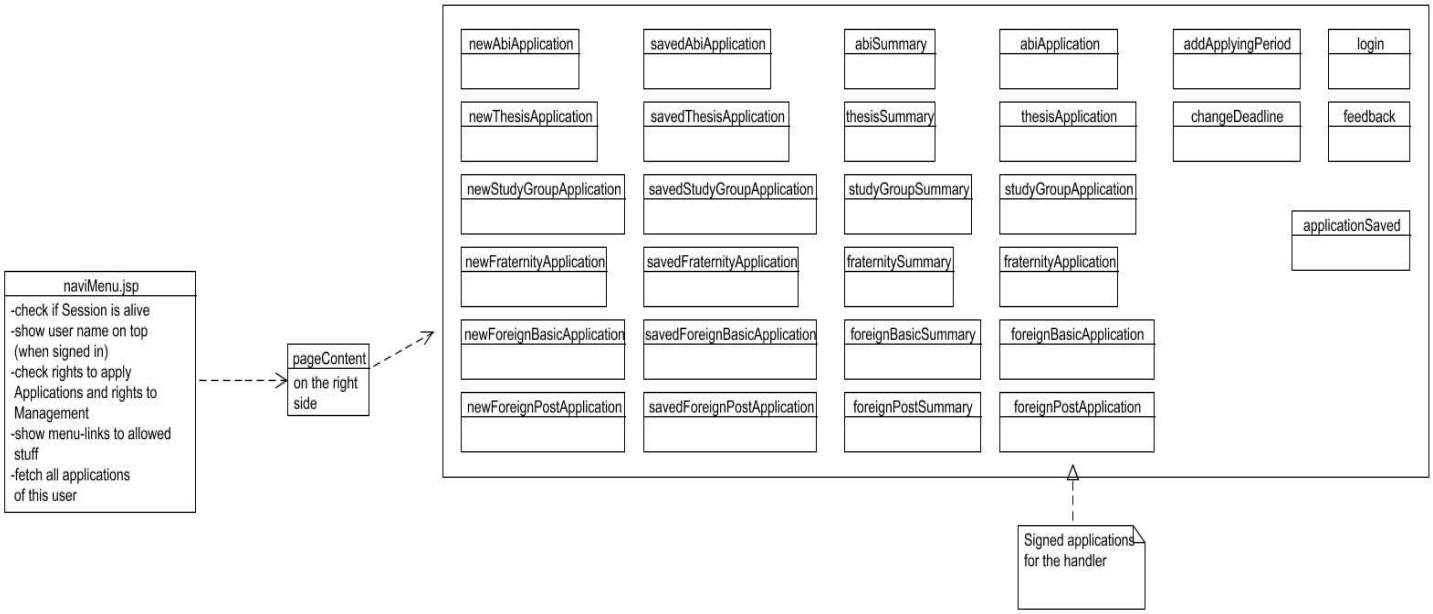
5 Osajärjestelmä: Servlet-kaavio

-Servlet-kaavio jäi vain servlet-listaukseksi ajan puutteen takia.

5.1Viestinvälitysrajapinta

-Luokkakaavioiden oliot tarjoavat get- ja set-metodit tiedon välitykseen ja tallennukseen. Lisäksi Manager-luokka tarjoaa saman rajapinnan metodeillaan, jotka käsittelevät tietokantatasoa.

6 Osajärjestelmä: Jsp-kaavio



6.1Viestinvälitysrajapinta

-Servletit tarjoavat get- ja set-metodit jsp-luokille.

7 Tieto-oliot

- *Session-luokka*

8 Viestinvälitysoliot

- *Manager-luokka*

9 Apuoliot

- *Util-luokka*

10 Tietokanta

Boolean-tyypin toteutukseen käytettiin CHAR(1)-tyyppiä.
Create table-lauseet:

```
CREATE TABLE UserPage (
UnixID VARCHAR2(9) NOT NULL,
LastPageVisited VARCHAR2(100),
PRIMARY KEY (UnixID)
);

CREATE TABLE UserRightsAndStuff (
UserRightsID INTEGER NOT NULL,
UnixID VARCHAR2(9) NOT NULL,
Abi CHAR(1) CHECK (Abi IN (0,1)) NOT NULL,
Thesis CHAR(1) CHECK (Thesis IN (0,1)) NOT NULL,
StudyGroup CHAR(1) CHECK (StudyGroup IN (0,1)) NOT NULL,
Fraternity CHAR(1) CHECK (Fraternity IN (0,1)) NOT NULL,
Foreign CHAR(1) CHECK (Foreign IN (0,1)) NOT NULL,
PRIMARY KEY (UserRightsID, UnixID)
);

CREATE TABLE ViewAbi (
ViewAbiID INTEGER NOT NULL,
UnixID VARCHAR2(9) NOT NULL,
Attendees CHAR(1) CHECK (Attendees IN (0,1)) NOT NULL,
Places CHAR(1) CHECK (Places IN (0,1)) NOT NULL,
ReportGotten CHAR(1) CHECK (ReportGotten IN (0,1)) NOT NULL,
OtherGrants CHAR(1) CHECK (OtherGrants IN (0,1)) NOT NULL,
AppliedPerAttendee CHAR(1) CHECK (AppliedPerAttendee IN (0,1)) NOT NULL,
PRIMARY KEY (ViewAbiID, UnixID)
);


CREATE TABLE ViewThesis (
ViewThesisID INTEGER NOT NULL,
UnixID VARCHAR2(9) NOT NULL,
ThesisTimeDone CHAR(1) CHECK (ThesisTimeDone IN (0,1)) NOT NULL,
ThesisTimeLeft CHAR(1) CHECK (ThesisTimeLeft IN (0,1)) NOT NULL,
Stations CHAR(1) CHECK (Stations IN (0,1)) NOT NULL,
WishedWeeks CHAR(1) CHECK (WishedWeeks IN (0,1)) NOT NULL,
PRIMARY KEY (ViewThesisID, UnixID)
);
```

```
CREATE TABLE ViewStudyGroup (
ViewStudyGroupID INTEGER NOT NULL,
UnixID VARCHAR2(9) NOT NULL,
FieldOfScience CHAR(1) CHECK (FieldOfScience IN (0,1)) NOT NULL,
EstimatedStudentCount CHAR(1) CHECK (EstimatedStudentCount IN (0,1)) NOT NULL,
ReportGotten CHAR(1) CHECK (ReportGotten IN (0,1)) NOT NULL,
HoursTotal CHAR(1) CHECK (HoursTotal IN (0,1)) NOT NULL,
TeachingType CHAR(1) CHECK (TeachingType IN (0,1)) NOT NULL,
SalaryTotal CHAR(1) CHECK (SalaryTotal IN (0,1)) NOT NULL,
PRIMARY KEY (ViewStudyGroupID, UnixID)
);

CREATE TABLE ViewFraternity (
ViewFraternityID INTEGER NOT NULL,
UnixID VARCHAR2(9) NOT NULL,
OrgName CHAR(1) CHECK (OrgName IN (0,1)) NOT NULL,
OrgFieldOfScience CHAR(1) CHECK (OrgFieldOfScience IN (0,1)) NOT NULL,
RelationToStudies CHAR(1) CHECK (RelationToStudies IN (0,1)) NOT NULL,
Turnout CHAR(1) CHECK (Turnout IN (0,1)) NOT NULL,
OtherGrants CHAR(1) CHECK (OtherGrants IN (0,1)) NOT NULL,
AppliedPerAttendee CHAR(1) CHECK (AppliedPerAttendee IN (0,1)) NOT NULL,
EstimatedExpense CHAR(1) CHECK (EstimatedExpense IN (0,1)) NOT NULL,
AppliedAmount CHAR(1) CHECK (AppliedAmount IN (0,1)) NOT NULL,
PRIMARY KEY (ViewFraternityID, UnixID)
);

CREATE TABLE ViewForeignBasic (
ViewForeignBasicID INTEGER NOT NULL,
UnixID VARCHAR2(9) NOT NULL,
Attendee CHAR(1) CHECK (Attendee IN (0,1)) NOT NULL,
Nationality CHAR(1) CHECK (Nationality IN (0,1)) NOT NULL,
CreditUnitsLastYear CHAR(1) CHECK (CreditUnitsLastYear IN (0,1)) NOT NULL,
CreditUnitsEstimated CHAR(1) CHECK (CreditUnitsEstimated IN (0,1)) NOT NULL,
EstimatedGraduationTime CHAR(1) CHECK (EstimatedGraduationTime IN (0,1)) NOT NULL,
ThesisState CHAR(1) CHECK (ThesisState IN (0,1)) NOT NULL,
ThesisTimeLeft CHAR(1) CHECK (ThesisTimeLeft IN (0,1)) NOT NULL,
Recommendation1 CHAR(1) CHECK (Recommendation1 IN (0,1)) NOT NULL,
Recommendation2 CHAR(1) CHECK (Recommendation2 IN (0,1)) NOT NULL,
PRIMARY KEY (ViewForeignBasicID, UnixID)
);
```

```
CREATE TABLE ViewForeignPost (
ViewForeignPostID INTEGER NOT NULL,
UnixID VARCHAR2(9) NOT NULL,
Attendee CHAR(1) CHECK (Attendee IN (0,1)) NOT NULL,
Nationality CHAR(1) CHECK (Nationality IN (0,1)) NOT NULL,
CreditUnitsLastYear CHAR(1) CHECK (CreditUnitsLastYear IN (0,1)) NOT NULL,
TakesPostGraduateDegree CHAR(1) CHECK (TakesPostGraduateDegree IN (0,1)) NOT NULL,
DegreeLeft CHAR(1) CHECK (DegreeLeft IN (0,1)) NOT NULL,
ResearchDone CHAR(1) CHECK (ResearchDone IN (0,1)) NOT NULL,
Recommendation1 CHAR(1) CHECK (Recommendation1 IN (0,1)) NOT NULL,
Recommendation2 CHAR(1) CHECK (Recommendation2 IN (0,1)) NOT NULL,
PRIMARY KEY (ViewForeignPostID, UnixID)
);

CREATE TABLE Scholarship (
ScholarshipID INTEGER NOT NULL,
Name VARCHAR2(100) NOT NULL,
ForBasicStudent CHAR(1) CHECK (ForBasicStudent IN (0,1)) NOT NULL,
ForPostStudent CHAR(1) CHECK (ForPostStudent IN (0,1)) NOT NULL,
CanApplyManyAtOneTime CHAR(1) CHECK (CanApplyManyAtOneTime IN (0,1)) NOT NULL,
CanGetMany CHAR(1) CHECK (CanGetMany IN (0,1)) NOT NULL,
HasRecommendations CHAR(1) CHECK (HasRecommendations IN (0,1)) NOT NULL,
PRIMARY KEY (ScholarshipID)
);

CREATE TABLE ApplyingPeriod (
ApplyingPeriodID INTEGER NOT NULL,
ScholarshipID INTEGER NOT NULL,
BeginDate DATE NOT NULL,
EndDate DATE NOT NULL,
HasRecommendationsOn CHAR(1) CHECK (HasRecommendationsOn IN (0,1)) NOT NULL,
PRIMARY KEY (ApplyingPeriodID, ScholarshipID),
FOREIGN KEY (ScholarshipID)
 REFERENCES Scholarship
);


CREATE TABLE ApplicationState (
ApplicationStateID INTEGER NOT NULL,
Name VARCHAR2(40) NOT NULL,
PRIMARY KEY (ApplicationStateID)
);
```

```
CREATE TABLE BiggestApplicationID (
SingletonID INTEGER NOT NULL,
BiggestID INTEGER NOT NULL
 CHECK (BiggestID > -1),
PRIMARY KEY (SingletonID)
);


CREATE TABLE Filler (
ApplicationID INTEGER NOT NULL,                    /*FillerID:itä on monta samaa, mutta
                                                   ApplicationID:itä vain yksi samaa.*/
FillerID INTEGER NOT NULL,
FillerUnixID VARCHAR2(9) NOT NULL,
FillerStudentID VARCHAR2(20) NOT NULL,
FillerPostGraduate CHAR(1) CHECK (FillerPostGraduate IN (0,1)) NOT NULL,
FillerName VARCHAR2(100) NOT NULL,
FillerEmail VARCHAR2(100),
FillerPhoneNo VARCHAR2(30),
FillerStreetAddress VARCHAR2(100),
FillerPostNo VARCHAR2(10),
FillerPostOffice VARCHAR2(30),
FillerNationality VARCHAR2(50),
PRIMARY KEY (ApplicationID)
);

CREATE TABLE ApplicationAbi (
ApplicationAbiID INTEGER NOT NULL,
ScholarshipID INTEGER NOT NULL,
ApplyingPeriodID INTEGER NOT NULL,
ApplicationStateID INTEGER NOT NULL,
Signed CHAR(1) CHECK (Signed IN (0,1)) NOT NULL,
SignedDate DATE,
DeletedBy VARCHAR2(9),
DeletedDate DATE,
HolderAmount INTEGER,                    /*tästä alkaa abiin liittyvät tiedot*/
FieldsOfStudy VARCHAR2(250),
ReasonsForAbiAmount VARCHAR2(1000),
EventDescription VARCHAR2(1000),
OtherGrants INTEGER,
ApplicantName VARCHAR2(100),
ApplicantAccountNo VARCHAR2(30),
CONSTRAINT ApplicationAbi_pkey
```

```
 PRIMARY KEY (ApplicationAbiID),
CONSTRAINT ApplicationAbi_f1
 FOREIGN KEY (ScholarshipID)
  REFERENCES Scholarship,
CONSTRAINT ApplicationAbi_f2
 FOREIGN KEY (ApplyingPeriodID)
  REFERENCES ApplyingPeriod,
CONSTRAINT ApplicationAbi_f3
 FOREIGN KEY (ApplicationStateID)
  REFERENCES ApplicationState
);

CREATE TABLE Abi_info (
Abi_infoID INTEGER NOT NULL,
ApplicationAbiID INTEGER NOT NULL,
Municipality VARCHAR2(50) NOT NULL,
Event VARCHAR2(100),
AbiAmount INTEGER,
PRIMARY KEY (Abi_infoID),
FOREIGN KEY (ApplicationAbiID)
 REFERENCES ApplicationAbi
);

CREATE TABLE RealizedAbi_info (
RealizedAbi_infoID INTEGER NOT NULL,
ApplicationAbiID INTEGER NOT NULL,
Municipality VARCHAR2(50) NOT NULL,
Event VARCHAR2(100),
AbiAmount INTEGER,
PRIMARY KEY (RealizedAbi_infoID),
FOREIGN KEY (ApplicationAbiID)
 REFERENCES ApplicationAbi
);
```

```
CREATE TABLE Expenses (
ExpensesID INTEGER NOT NULL,
ApplicationID INTEGER NOT NULL,
Transport INTEGER
 CHECK (Transport > -1),
TransportDescription VARCHAR2(100),
Accommodation INTEGER
 CHECK (Accommodation > -1),
AccommodationDescription VARCHAR2(100),
Eating INTEGER
 CHECK (Eating > -1),
EatingDescription VARCHAR2(100),
Other INTEGER
 CHECK (Other > -1),
OtherDescription VARCHAR2(100),
PRIMARY KEY (ExpensesID),
FOREIGN KEY (ApplicationID)
 REFERENCES ApplicationAbi,
FOREIGN KEY (ApplicationID)                    /*ID CAN REFER TO 2 TABLES*/
 REFERENCES ApplicationFraternity
);

CREATE TABLE RealizedExpenses (
RealizedExpensesID INTEGER NOT NULL,
ApplicationID INTEGER NOT NULL,
Transport INTEGER
 CHECK (Transport > -1),
TransportDescription VARCHAR2(100),
Accommodation INTEGER
 CHECK (Accommodation > -1),
AccommodationDescription VARCHAR2(100),
Eating INTEGER
 CHECK (Eating > -1),
EatingDescription VARCHAR2(100),
Other INTEGER
 CHECK (Other > -1),
OtherDescription VARCHAR2(100),
PRIMARY KEY (RealizedExpensesID),
FOREIGN KEY (ApplicationID)
 REFERENCES ApplicationAbi,
FOREIGN KEY (ApplicationID)                    /*ID CAN REFER TO 2 TABLES*/
```

```
 REFERENCES ApplicationFraternity
);



CREATE TABLE ApplicationThesis (
ApplicationThesisID INTEGER NOT NULL,
ScholarshipID INTEGER NOT NULL,
ApplyingPeriodID INTEGER NOT NULL,
ApplicationStateID INTEGER NOT NULL,
Signed CHAR(1) CHECK (Signed IN (0,1)) NOT NULL,
SignedDate DATE,
DeletedBy VARCHAR2(9),
DeletedDate DATE,
Subject VARCHAR2(200),          /*tästä alkaa thesis-tyypin omat tiedot*/
BeginDate DATE,
EndDate DATE,
ShortDescription VARCHAR2(1000),
ReasonsForGrant VARCHAR2(1000),
WishedDate DATE,
WishedLength INT,
ReasonsForStations VARCHAR2(1000),
Station1 INTEGER,
Station2 INTEGER,
Station3 INTEGER,
Station4 INTEGER,
CONSTRAINT ApplicationThesis_pkey
 PRIMARY KEY (ApplicationThesisID),
CONSTRAINT ApplicationThesis_f1
 FOREIGN KEY (ScholarshipID)
  REFERENCES Scholarship,
CONSTRAINT ApplicationThesis_f2
 FOREIGN KEY (ApplyingPeriodID)
  REFERENCES ApplyingPeriod,
CONSTRAINT ApplicationThesis_f3
 FOREIGN KEY (ApplicationStateID)
  REFERENCES ApplicationState
);
```

```
CREATE TABLE Stations (
StationsID INTEGER NOT NULL,
Name VARCHAR2(100) NOT NULL,
PRIMARY KEY (StationsID)
);

CREATE TABLE RecommendationThesis (
RecommendationID VARCHAR2(100) NOT NULL,
IsValid CHAR(1) CHECK (IsValid IN (0,1)),
ApplicationThesisID INTEGER NOT NULL,
RecommenderName VARCHAR2(100),
ApplierGoodForGrant CHAR(1) CHECK (ApplierGoodForGrant IN (0,1)),
ReasonsForYes VARCHAR2(500),
RecommendationSigned CHAR(1) CHECK (RecommendationSigned IN (0,1)),
PRIMARY KEY (RecommendationID),
FOREIGN KEY (ApplicationThesisID)
 REFERENCES ApplicationThesis
);

CREATE TABLE ReportThesis (
ReportThesisID INTEGER NOT NULL,
ApplicationThesisID INTEGER NOT NULL,
GrantUsedDate DATE,
DescriptionOfWork VARCHAR2(1000),
PeriodSignificanceForThesis VARCHAR2(1000),
SuggestionsForDevelopment VARCHAR2(1000),
PRIMARY KEY (ReportThesisID),
FOREIGN KEY (ApplicationThesisID)
 REFERENCES ApplicationThesis
);
```

```
CREATE TABLE ApplicationStudyGroup (
ApplicationStudyGroupID INTEGER NOT NULL,
ScholarshipID INTEGER NOT NULL,
ApplyingPeriodID INTEGER NOT NULL,
ApplicationStateID INTEGER NOT NULL,
Signed CHAR(1) CHECK (Signed IN (0,1)) NOT NULL,
SignedDate DATE,
DeletedBy VARCHAR2(9),
DeletedDate DATE,
ApplicantName VARCHAR2(100),                    /*tästä alkaa StudyGroup-tyypin omat tiedot*/
ApplicantAccountNo VARCHAR2(50),
Subject VARCHAR2(100),
FieldOfScience VARCHAR2(100),
ReasonsForStudyGroup VARCHAR2(1000),
AttendeeAmount INTEGER,
ReasonsForAmount VARCHAR2(200),
FreeText VARCHAR2 (500),
CONSTRAINT ApplicationStudyGroup_pkey
 PRIMARY KEY (ApplicationStudyGroupID),
CONSTRAINT ApplicationStudyGroup_f1
 FOREIGN KEY (ScholarshipID)
  REFERENCES Scholarship,
CONSTRAINT ApplicationStudyGroup_f2
 FOREIGN KEY (ApplyingPeriodID)
  REFERENCES ApplyingPeriod,
CONSTRAINT ApplicationStudyGroup_f3
 FOREIGN KEY (ApplicationStateID)
  REFERENCES ApplicationState
);
CREATE TABLE TeachingHours (
TeachingHoursID INTEGER NOT NULL,
ApplicationStudyGroupID INTEGER NOT NULL,
Pro INTEGER,
Dos INTEGER,
Toh INTEGER,
Lis INTEGER,
HighDegree INTEGER,
LowDegree INTEGER,
PRIMARY KEY (TeachingHoursID),
FOREIGN KEY (ApplicationStudyGroupID)
 REFERENCES ApplicationStudyGroup
);
```

```
CREATE TABLE TeachingSubjects (
TeachingSubjectsID INTEGER NOT NULL,
ApplicationStudyGroupID INTEGER NOT NULL,
Subject VARCHAR2(100),
Hours INTEGER,
PRIMARY KEY (TeachingSubjectsID),
FOREIGN KEY (ApplicationStudyGroupID)
 REFERENCES ApplicationStudyGroup
);

CREATE TABLE RealizedTeachingHours (
RealizedTeachingHoursID INTEGER NOT NULL,
ApplicationStudyGroupID INTEGER NOT NULL,
Pro INTEGER,
Dos INTEGER,
Toh INTEGER,
Lis INTEGER,
HighDegree INTEGER,
LowDegree INTEGER,
PRIMARY KEY (RealizedTeachingHoursID),
FOREIGN KEY (ApplicationStudyGroupID)
 REFERENCES ApplicationStudyGroup
);

CREATE TABLE ReportStudyGroup (
ReportStudyGroupID INTEGER NOT NULL,
ApplicationStudyGroupID INTEGER NOT NULL,
BasicDegreeAmount INTEGER,
PostDegreeAmount INTEGER,
FieldOfScience VARCHAR2(100),
DescriptionOfStudying VARCHAR2(100),
StudyGroupSignificance VARCHAR2(1000),
TeachingFacilitys VARCHAR2(1000),
SuggestionsForDevelopment VARCHAR2(1000),
PRIMARY KEY (ReportStudyGroupID),
FOREIGN KEY (ApplicationStudyGroupID)
 REFERENCES ApplicationStudyGroup
);
```

```
CREATE TABLE ApplicationFraternity (
ApplicationFraternityID INTEGER NOT NULL,
ScholarshipID INTEGER NOT NULL,
ApplyingPeriodID INTEGER NOT NULL,
ApplicationStateID INTEGER NOT NULL,
Signed CHAR(1) CHECK (Signed IN (0,1)) NOT NULL,
SignedDate DATE,
DeletedBy VARCHAR2(9),
DeletedDate DATE,
ProjectName VARCHAR2(100),                    /*tästä alkaa Fraternity-tyypin omat tiedot*/
RelationToStudies VARCHAR2(1000),
ProjectGoals VARCHAR2(1000),
Schedule VARCHAR2(1000),
AttendeeAmount INTEGER,
OtherGrants INTEGER,
CONSTRAINT ApplicationFraternity_pkey
 PRIMARY KEY (ApplicationFraternityID),
CONSTRAINT ApplicationFraternity_f1
 FOREIGN KEY (ScholarshipID)
  REFERENCES Scholarship,
CONSTRAINT ApplicationFraternity_f2
 FOREIGN KEY (ApplyingPeriodID)
  REFERENCES ApplyingPeriod,
CONSTRAINT ApplicationFraternity_f3
 FOREIGN KEY (ApplicationStateID)
  REFERENCES ApplicationState
);
```