University of British Columbia, Vancouver

Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 2

Date: October 20, 2023

Group Number: 45

Name	Student #	CS ID	Email Address
Michael Perkins	35844802	mperki02	michael.alexander.perkins@gmail.com
Siddharth Nand	76648070	sidnand	snand233@gmail.com
Saif Karnawi	92839034	sfkrnwi	saifkarnawi2000@gmail.com

2. A brief (~2-3 sentences) summary of your project. Many of your TAs are managing multiple projects so this will help them remember details about your project.

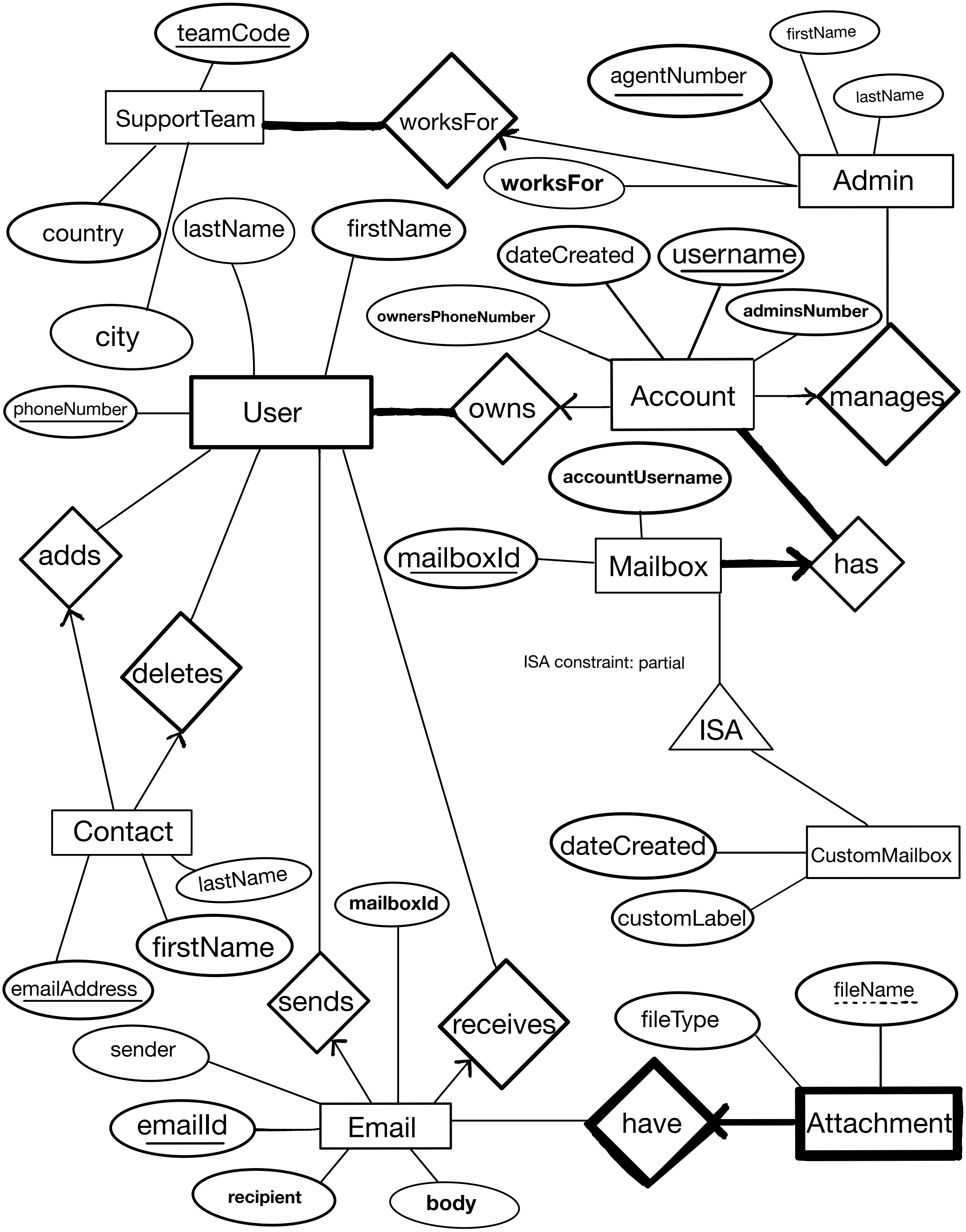
Our project is a database for an email service. Our database holds all the information that this service requires and would need access to in order to function correctly. It will function similarly to the large email providers such as Gmail.

3. The ER diagram you are basing your item #3 (below) on.

Changes to ER diagram:

- Added a “body” attribute to email.
- The fullName attribute for User, Contact, and Admin is now separated into two attributes for each: firstName and lastName.
- Removed “location” attribute from SupportTeam and added “Country” and “City” attributes. This is so that we can have other FDs that are not CKs or PKs.
- Added “sender” and “recipient” attributes to email to keep our ER diagram consistent with our table. Our table will be using these two attributes as foreign keys.
- Added an “ownersPhoneNumber” attribute to Account and a “worksFor” attribute to Admin. This is also to keep our ER diagrams consistent with our tables. “ownersPhoneNumber” and “worksFor” will both be foreign keys in the table.
- Added AdminsNumber to the Account entity as it will be used as a foreign key to reference the Admin.
- Removed emailAddress attribute from Mailbox and added an accountUsername attribute that will be used as a foreign key to reference account.
- Added mailboxId to email as a foreign key that references Mailbox.

Please see the next page.


teamCode
SupportTeam
worksFor
agentNumber
firstName
lastName
Admin
worksFor
country
lastName
firstName
city
dateCreated
username
adminsNumber
ownersPhoneNumber
phoneNumber
User
owns
Account
manages
accountUsername
mailboxId
Mailbox
has
adds
deletes
ISA constraint: partial
ISA
Contact
lastName
firstName
emailAddress
dateCreated
CustomMailbox
customLabel
mailboxId
sends
receives
fileName
fileType
sender
emailId
Email
have
Attachment
recipient
body

4. The schema derived from your ER diagram (above)

Email (
emailID INTEGER NOT NULL,
sender CHAR[20] NOT NULL,
recepient CHAR[20] NOT NULL,
mailboxId CHAR[20] NOT NULL,
body CHAR[200] NOT NULL)
Candidate keys: emailId. Primary key: emailId. Foreign keys: sender references account, recipient references account, mailboxId references mailbox.

Attachment (
fileName CHAR[20] NOT NULL,
emailId INTEGER NOT NULL,
fileType CHAR[20] NOT NULL)
Candidate keys: fileName, emailId. Primary key: fileName, emailId. Foreign keys: emailId references Email.

User (
phoneNumber CHAR[14] NOT NULL,
firstName CHAR[20] NOT NULL,
lastName CHAR[20] NOT NULL)
Candidate keys: phoneNumber. Primary key: phoneNumber. Foreign keys: none.

Account (
username CHAR[20] NOT NULL,
dateCreated DATE NOT NULL,
ownersPhoneNumber CHAR[14] NOT NULL,
adminsNumber INT NOT NULL)
Candidate keys: username. Primary key: username. Foreign keys: ownersPhoneNumber references User, adminsNumber references Admin.

Contact (
emailAddress CHAR[20] NOT NULL,
firstName CHAR[20] NOT NULL,
lastName CHAR[20] NOT NULL)
Candidate keys: emailAddress. Primary key: emailAddress. Foreign keys: none.

UserContacts(
userContactId INTEGER NOT NULL,
usersPhoneNumber CHAR[14] NOT NULL,
contactsEmail CHAR[20] NOT NULL)
Candidate keys: userContactId. Primary key: userContactId. Foreign keys: user references User, contactsEmail references Contact

Mailbox (
mailboxId CHAR[20] NOT NULL,
accountUsername CHAR[20] NOT NULL)
Candidate keys: mailboxId. Primary key: mailboxId. Foreign keys: accountUsername references Account.

CustomMailbox (
mailboxId CHAR[20] NOT NULL,
accountUsername CHAR[20] NOT NULL,
dateCreated DATE NOT NULL,
customLabel CHAR[20] NOT NULL)
Candidate keys: mailboxId. Primary key: mailboxId. Foreign keys: mailboxId references Mailbox, accountUsername references Account.

Admin (
agentNumber INT NOT NULL,
firstName CHAR[20] NOT NULL,
lastName CHAR[20] NOT NULL,
worksFor INT NOT NULL)
Candidate keys: agentNumber. Primary key: agentNumber. Foreign keys: worksFor references SupportTeam.

SupportTeam (
teamCode INT NOT NULL,
country CHAR[20] NOT NULL,
city CHAR[20] NOT NULL)
Candidate keys: teamCode. Primary key: teamCode. Foreign keys: none.

5. Functional Dependencies (FDs)

Email (emailID**, mailboxId,** sender, recipient, body)
FDs: emailID → mailboxId**,** sender, recipient, body

Attachment (fileName, **emailID references email**, fileType)
FDs: fileName, emailID → fileType

Account (username, **ownersPhoneNumber references user**, **adminsNumber references,** dateCreated)
FDs: username → ownersPhoneNumber, adminsNumber, dateCreated

Admin (agentNumber, firstName, lastName, **worksFor**)
FDs: agentNumber → firstName, lastName, worksFor

User (phoneNumber, firstName, lastName)
FDs: phoneNumber → firstName, lastName

Mailbox (mailboxID, **accountUsername references Account**)
FDs: mailboxID → accountUsername

CustomMailbox (**mailboxID references Mailbox**, **accountUsername references Account**, dateCreated, customLabel)
FDs: mailboxID → accountUsername, dateCreated, customLabel

Contact (emailAddress, firstName, lastName)
FDs: emailAddress → emailAddress, firstName, lastName

SupportTeam (teamCode, country, city)
FDs: teamCode → country, city
country → city

UserContacts (userContactId, **usersPhoneNumber references user, contactsEmail references contact**)
FDs: userContactId → user, contact

6. Normalization

Email (emailID**, mailboxId,** sender, recipient, body)
FDs: emailID → mailboxID, sender, recipient, body
This table is already in BCNF. There are no non-trivial relationships where the LFS is not a superkey.

Attachment (fileName, **emailID references email**, fileType)
FDs: fileName, emailID → fileType
This table is already in BCNF. There are no non-trivial relationships where the LFS is not a superkey.

Account (username, **ownersPhoneNumber references user**, **adminsNumber references Admin**, dateCreated)
FDs: username → dateCreated, ownersPhoneNumber
This table is already in BCNF. There are no non-trivial relationships where the LFS is not a superkey.

User (phoneNumber, firstName, lastName)
FDs: phoneNumber → firstName, lastName
This table is already in BCNF. There are no non-trivial relationships where the LFS is not a superkey.

Mailbox (mailboxID, **accountUsername references Account**)
FDs: mailboxID → accountUsername
This table is already in BCNF. There are no non-trivial relationships where the LFS is not a superkey.

CustomMailbox (**mailboxID**, **accountUsername**, dateCreated, customLabel)
FDs: mailboxID → emailAddress, accountUsername**,** dateCreated, customLabel
This table is already in BCNF. There are no non-trivial relationships where the LFS is not a superkey.

Contact (emailAddress, firstName, lastName)
FDs: emailAddress → emailAddress, firstName, lastName
This table is already in BCNF. There are no non-trivial relationships where the LFS is not a superkey.

Admin (agentNumber, **worksFor**, firstName, lastName)
FDs: agentNumber → firstName, lastNam, worksFor
This table is already in BCNF. There are no non-trivial relationships where the LFS is not a superkey.

SupportTeam (teamCode, country, city)
FDs: teamCode → country, city
country → city
We can see that we have country → city and country is not a super key. Let's normalize: (teamcode (country) city). We end up with R1(teamCode, country) and R2(country, city)

UserContacts (userContactId, **user references user, contact references contact**)
FDs: userContactId → user, contact
This table is already in BCNF. There are no non-trivial relationships where the LFS is not a superkey.

7. SQL DDL Statements

```
CREATE TABLE SupportTeam (

    teamCode INT,
    country VARCHAR(20) NOT NULL,
    city VARCHAR(20) NOT NULL,

    PRIMARY KEY (teamCode)

);

CREATE TABLE "User" (

    phoneNumber VARCHAR(14),
    firstName VARCHAR(20) NOT NULL,
    lastName VARCHAR(20) NOT NULL,
```

```sql
    PRIMARY KEY (phoneNumber)

);


CREATE TABLE Admin (

    agentNumber INT,
    firstName VARCHAR(20) NOT NULL,
    lastName VARCHAR(20) NOT NULL,
    worksFor INT NOT NULL,

    PRIMARY KEY (agentNumber),
    FOREIGN KEY (worksFor) REFERENCES SupportTeam(teamCode) ON DELETE
CASCADE

);

CREATE TABLE Account (

    username VARCHAR(20),
    dateCreated DATE NOT NULL,
    ownersPhoneNumber VARCHAR(14) NOT NULL,
    adminsNumber INT NOT NULL,

    PRIMARY KEY (username),
    FOREIGN KEY (ownersPhoneNumber) REFERENCES "User"(phoneNumber) ON
DELETE CASCADE,
    FOREIGN KEY (adminsNumber) REFERENCES Admin(agentNumber) ON DELETE
CASCADE

);

CREATE TABLE Mailbox (

    mailboxID INT,
    accountUsername VARCHAR(20) NOT NULL,

    PRIMARY KEY (mailboxID),
    FOREIGN KEY (accountUsername) REFERENCES Account(username) ON DELETE
CASCADE
```

```
);




CREATE TABLE CustomMailbox (

    mailboxID INT,
    accountUsername VARCHAR(20) NOT NULL,
    dateCreated DATE NOT NULL,
    customLabel VARCHAR(20) NOT NULL,

    PRIMARY KEY (mailboxID),
    FOREIGN KEY (accountUsername) REFERENCES Account(username) ON DELETE
CASCADE

);

CREATE TABLE Email (

    emailID INT,
    sender VARCHAR(20) NOT NULL,
    recipient VARCHAR(20) NOT NULL,
    body VARCHAR(200) NOT NULL,
    mailboxID INT NOT NULL,

    PRIMARY KEY (emailID),
    FOREIGN KEY (sender) REFERENCES Account(username) ON DELETE CASCADE,
    FOREIGN KEY (recipient) REFERENCES Account(username) ON DELETE CASCADE,
    FOREIGN KEY (mailboxID) REFERENCES Mailbox(mailboxID) ON DELETE CASCADE

);

CREATE TABLE Attachment (

    fileName VARCHAR(20),
    emailID INT NOT NULL,
    fileType VARCHAR(20) NOT NULL,

    PRIMARY KEY (fileName, emailID),
    FOREIGN KEY (emailID) REFERENCES Email(emailID) ON DELETE CASCADE
```

```
);




CREATE TABLE Contact (

    emailAddress VARCHAR(20),
    firstName VARCHAR(20) NOT NULL,
    lastName VARCHAR(20) NOT NULL,

    PRIMARY KEY (emailAddress)

);

CREATE TABLE UserContacts (

    userContactID INT,
    usersPhoneNumber VARCHAR(14) NOT NULL,
    contactsEmailAddress VARCHAR(20) NOT NULL,

    PRIMARY KEY (userContactID),
    FOREIGN KEY (usersPhoneNumber) REFERENCES "User"(phoneNumber) ON DELETE
CASCADE,
    FOREIGN KEY (contactsEmailAddress) REFERENCES Contact(emailAddress) ON
DELETE CASCADE

);
```

8. Populating Tables

```
INSERT INTO SupportTeam VALUES (1, 'Canada', 'Toronto');
INSERT INTO SupportTeam VALUES (2, 'Canada', 'Vancouver');
INSERT INTO SupportTeam VALUES (3, 'Canada', 'Montreal');
INSERT INTO SupportTeam VALUES (4, 'United States', 'New York');
INSERT INTO SupportTeam VALUES (5, 'United States', 'Los Angeles');

INSERT INTO "User" VALUES ('+17786652266', 'Siddharth', 'Nand');
INSERT INTO "User" VALUES ('+34432348866', 'Michael', 'Perkins');
INSERT INTO "User" VALUES ('+03441128899', 'Saif', 'Karnawi');
INSERT INTO "User" VALUES ('+15557771111', 'Jennifer', 'Smith');
INSERT INTO "User" VALUES ('+26668882222', 'Lora', 'Hill');

INSERT INTO Admin VALUES (1010, 'Dave', 'Colby', 1);
INSERT INTO Admin VALUES (1011, 'Emma', 'Stome', 1);
INSERT INTO Admin VALUES (1012, 'Jessica', 'Khali', 2);
INSERT INTO Admin VALUES (1013, 'Jone', 'Doe', 3);
INSERT INTO Admin VALUES (1014, 'Jane', 'Doe', 4);

INSERT INTO Account VALUES ('siddharthnand', TO_DATE('2019-01-01',
'YYYY-MM-DD'), '+17786652266', 1010);
INSERT INTO Account VALUES ('michaelperkins', TO_DATE('2019-01-01',
'YYYY-MM-DD'), '+34432348866', 1011);
INSERT INTO Account VALUES ('saifkarnawi', TO_DATE('2019-01-01',
'YYYY-MM-DD'), '+03441128899', 1012);
INSERT INTO Account VALUES ('jennifersmith', TO_DATE('2019-01-01',
'YYYY-MM-DD'), '+15557771111', 1013);
INSERT INTO Account VALUES ('lorahill', TO_DATE('2019-01-01',
'YYYY-MM-DD'), '+26668882222', 1014);
INSERT INTO Account VALUES ('k_smith', TO_DATE('2019-01-01', 'YYYY-MM-DD'),
'+26668882222', 1014);

INSERT INTO Mailbox VALUES (1, 'siddharthnand');
INSERT INTO Mailbox VALUES (2, 'michaelperkins');
INSERT INTO Mailbox VALUES (3, 'saifkarnawi');
INSERT INTO Mailbox VALUES (4, 'jennifersmith');
INSERT INTO Mailbox VALUES (5, 'lorahill');
INSERT INTO Mailbox VALUES (6, 'k_smith');

INSERT INTO CustomMailbox VALUES (1, 'siddharthnand', TO_DATE('2019-01-01',
'YYYY-MM-DD'), 'Important');
INSERT INTO CustomMailbox VALUES (2, 'michaelperkins',
TO_DATE('2019-01-01', 'YYYY-MM-DD'), 'Work');
INSERT INTO CustomMailbox VALUES (3, 'saifkarnawi', TO_DATE('2019-01-01',
'YYYY-MM-DD'), 'Family');
INSERT INTO CustomMailbox VALUES (4, 'jennifersmith', TO_DATE('2019-01-01',
'YYYY-MM-DD'), 'Friends');
INSERT INTO CustomMailbox VALUES (5, 'lorahill', TO_DATE('2019-01-01',
'YYYY-MM-DD'), 'Important');
INSERT INTO CustomMailbox VALUES (6, 'k_smith', TO_DATE('2019-01-01',
'YYYY-MM-DD'), 'Junk');

INSERT INTO Email VALUES (1, 'siddharthnand', 'michaelperkins', 'Hello, how
are you?', 1);
```

```
INSERT INTO Email VALUES (2, 'michaelperkins', 'siddharthnand', 'I am good,
how are you?', 2);
INSERT INTO Email VALUES (3, 'saifkarnawi', 'jennifersmith', 'Can we have a
meeting tomorrow at 5pm?', 3);
INSERT INTO Email VALUES (4, 'jennifersmith', 'saifkarnawi', 'Can we delay
it till next week?', 4);
INSERT INTO Email VALUES (5, 'lorahill', 'k_smith', 'I am going to be late
for dinner tonight', 5);
INSERT INTO Email VALUES (6, 'k_smith', 'lorahill', 'Ok, I will wait for
you', 6);
INSERT INTO Email VALUES (7, 'siddharthnand', 'lorahill', 'You did not get
the job offer', 1);

INSERT INTO Attachment VALUES ('attachment1', 1, 'pdf');
INSERT INTO Attachment VALUES ('attachment2', 2, 'doc');
INSERT INTO Attachment VALUES ('attachment3', 3, 'html');
INSERT INTO Attachment VALUES ('attachment4', 4, 'zip');
INSERT INTO Attachment VALUES ('attachment5', 5, 'pdf');

INSERT INTO Contact VALUES ('siddharthnand@mail.com', 'Siddharth', 'Nand');
INSERT INTO Contact VALUES ('michaelperkins@mail.com', 'Michael',
'Perkins');
INSERT INTO Contact VALUES ('saifkarnawi@mail.com', 'Saif', 'Karnawi');
INSERT INTO Contact VALUES ('nand_s@gmail.com', 'Siddharth', 'Nand');
INSERT INTO Contact VALUES ('jennifersmith@mail.com', 'Jennifer', 'Smith');
INSERT INTO Contact VALUES ('admin@ubc.ca', 'UBC', 'Admin');

INSERT INTO UserContacts VALUES (1, '+17786652266',
'michaelperkins@mail.com');
INSERT INTO UserContacts VALUES (2, '+17786652266',
'saifkarnawi@mail.com');
INSERT INTO UserContacts VALUES (3, '+17786652266', 'nand_s@gmail.com');
INSERT INTO UserContacts VALUES (4, '+34432348866', 'admin@ubc.ca');
INSERT INTO UserContacts VALUES (5, '+03441128899',
'jennifersmith@mail.com');
```