CENT Database Management System
and
Application

Submitted by

Shripad Laddha (711417912)
Prachi Sankpal(752791390)
Anupama Rao(809474520)

Table of index

Project Report

1. Project overview

The CENT researcher centre has a lot of physical equipment in the lab. This project focuses on which of the equipment are used as part of faculty-led research projects, student-led research projects or as part of classroom activities. In every case equipment must be loaned out and kept track of so that it does not go missing i.e. Inventory management of the products.

1.1 Project Requirements:

1. CENT must maintain an inventory of equipment in its lab. Some of the equipment is eligible for loan others are not.
2. All equipment, whether loanable or not should have an inventory bar-code readable tag in it which uniquely identifies the equipment.
3. Inventory should be classified by item type such as equipment manufacturer, make and model.
4. While each physical item should be inventoried, but there should be a report which can provide a count of how many items of a specific item type: manufacturer, make and model are available.
5. Researchers (students, faculty, etc.) should be able to loan out any loanable equipment by item type.
6. The loan process should work like a shopping cart where you can create a new loan and add multiple inventoried items of a specific type to the loan. This is called an Ad-hoc loan.
7. Ad-hoc loans can be renewed. Also, the return date can be set manually.
8. There needs to be a loan template facility so than common loan setups can be created. These are called activity loans. For example, for one class activity, a specific model of router and Wi-fi access point are required to be loaned for the activity. In this case a student can ask to loan the equipment required to complete Lab XYZ for course IST123 and then the loan shopping cart would automatically be loaded with specific items from inventory to be loaned.
9. All items on loan must have a return date. And a report should be able to be generated for items on loan and overdue items.
10. Return dates are customizable as part of the activity loan template.
11. The system should easily setup users using SUID, net ID or some other facility to guarantee the identity of the loaner.
12. Any activity loan template should have a point of contact so that a user is identified as responsible for the setup of the activity loans.

13. Activity loans should have fields to describe the nature and purpose of the loan or a least a link to a URL. So that users know what to do with the equipment.
14. All equipment on loan of the same type should have notes so that important information can be communicated regarding how to use and or setup the equipment.
15. For every loan, there should be a record of the user who processed the loan. This will typically be a user affiliated with CENT as a faculty member or technician.
16. There needs to be a list of people affiliated with CENT who are permitted to process loans.
17. Anyone affiliated with cent should get notifications when equipment is overdue.
18. The user loaning the equipment should get notifications when equipment is overdue.

1.2 Entities & Attributes:

User: username (unique, required), user id(required), first name(required),last name(required),borrow date(required), due date(required), reason(required), designation(required), barcode of the equipment borrowed(required, multivariable).

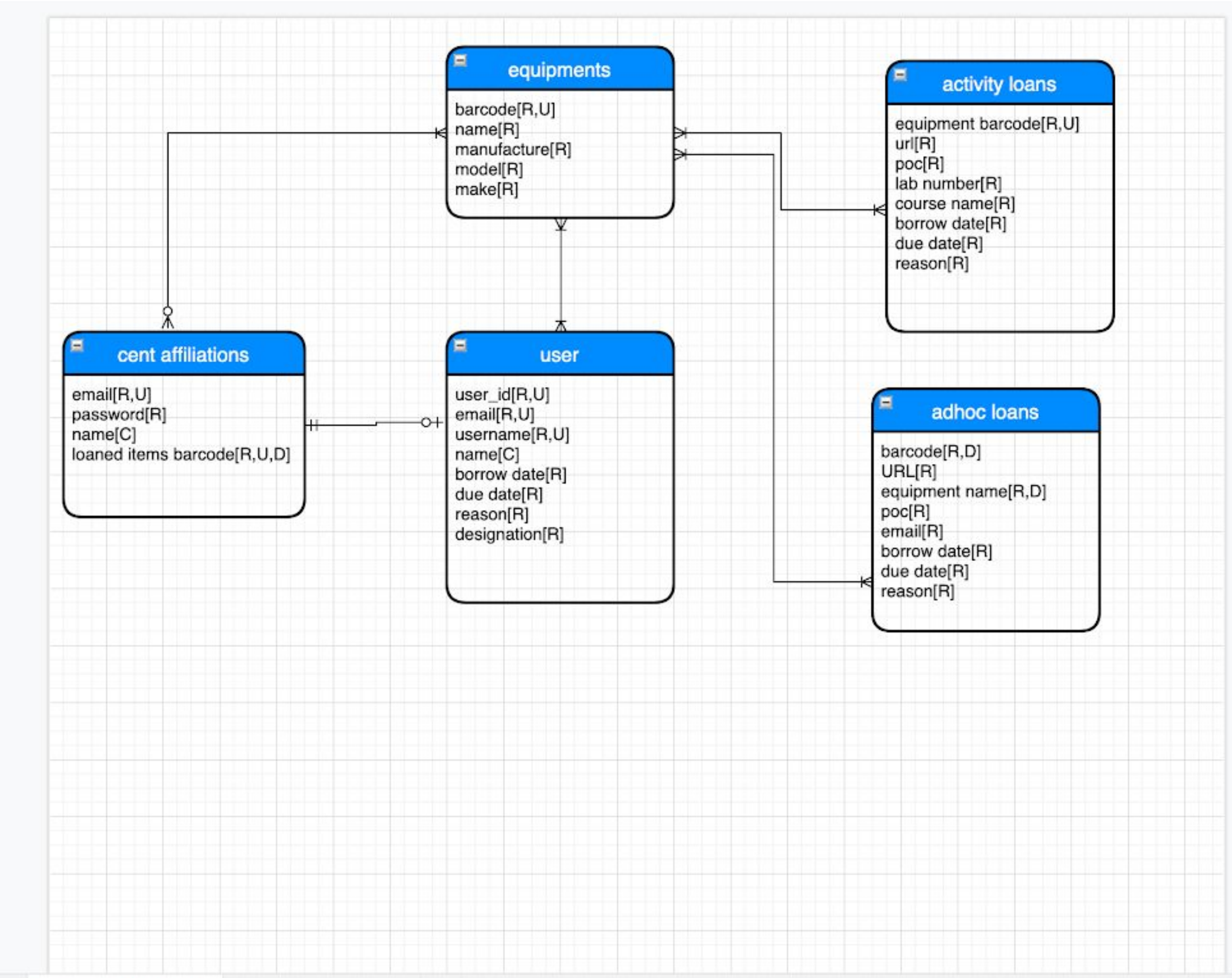
Equipment: name(required), manufacturer(required), model(required), make(required), barcode (required, unique).

Activity Loans: identity number(required), equipment barcode(required,multi variable), URL(required), point of contact(required),lab number(required), course name(required),course name(required),borrow date(required), due date(required), reason(required).

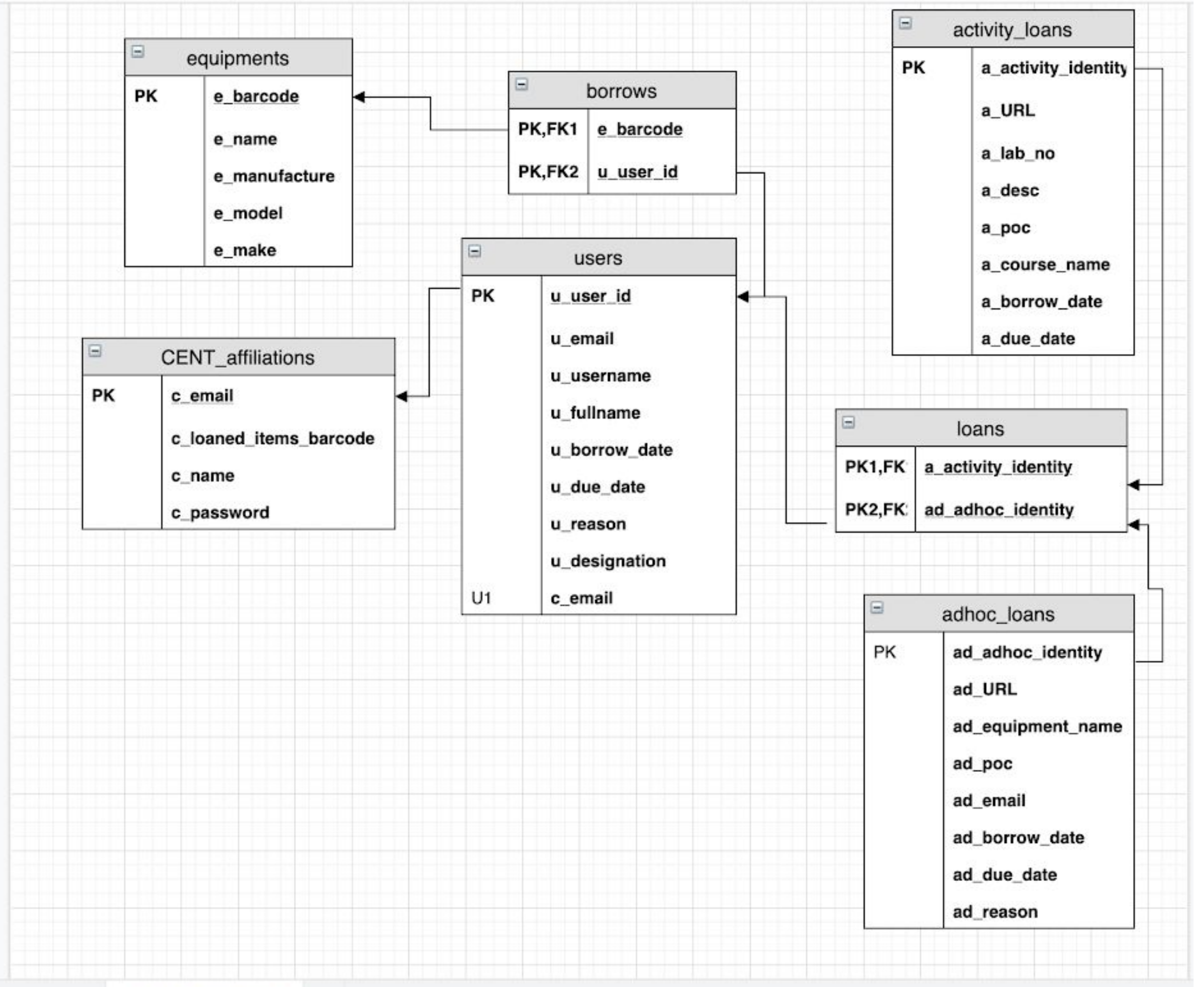
Ad-hoc Loans: identity number(required), equipment barcode(required), URL(required), point of contact(required),email(required), borrow date(required), due date(required), reason(required).

CENT affiliations: identity number(required), email(required),password(required),first name(required), last name(required), barcode of items loaned out(required).

2.Conceptual Diagram



3.Logical Diagram



3.1 Normalization

In our project, we do not need to normalize as it is a new database and we do not have existing data. We created our tables and inserted values in a way that it would satisfy normalization rules.

4.Script

```
--DROPPING CONSTRAINT QUERIES
IF EXISTS(SELECT * FROM INFORMATION_SCHEMA.TABLE_CONSTRAINTS WHERE
CONSTRAINT_NAME='fk_ad_barcode')
ALTER TABLE adhoc_loans
DROP
CONSTRAINT fk_ad_barcode

go
IF EXISTS(SELECT * FROM INFORMATION_SCHEMA.TABLE_CONSTRAINTS WHERE
CONSTRAINT_NAME='fk_activity_barcode')
ALTER TABLE activity_loans
DROP
CONSTRAINT fk_activity_barcode
go

IF EXISTS(SELECT * FROM INFORMATION_SCHEMA.TABLE_CONSTRAINTS WHERE
CONSTRAINT_NAME='fk_cent_barcode')
ALTER TABLE cent_affiliations
DROP
CONSTRAINT fk_cent_barcode
go

IF EXISTS(SELECT * FROM INFORMATION_SCHEMA.TABLE_CONSTRAINTS WHERE
CONSTRAINT_NAME='FK_user_barcode')
Alter table users
drop
constraint FK_user_barcode
go

IF EXISTS(SELECT * FROM INFORMATION_SCHEMA.TABLE_CONSTRAINTS WHERE
CONSTRAINT_NAME='PK_bridge_user_equipment')
Alter table bridge_user_equipment
drop
constraint PK_bridge_user_equipment
go

IF EXISTS(SELECT * FROM INFORMATION_SCHEMA.TABLE_CONSTRAINTS WHERE
CONSTRAINT_NAME='FK_bridge_user')
Alter table bridge_user_equipment
```

```
drop
constraint FK_bridge_user
go

IF EXISTS(SELECT * FROM INFORMATION_SCHEMA.TABLE_CONSTRAINTS WHERE
CONSTRAINT_NAME='FK_bridge_equipment')
Alter table bridge_user_equipment
drop
constraint FK_bridge_equipment
go

-- CREATING TABLES FOR CENT PROJECT

IF EXISTS(SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME='users')
DROP TABLE users
go
CREATE TABLE users (
   u_user_id int not null,
   u_email varchar(50) not null,
   u_username varchar(50) not null,
   u_password varchar(50) not null,
   u_firstname varchar(50) not null,
   u_lastname varchar(50) not null,
   u_borrow_date date not null,
   u_due_date date not null,
   u_reason varchar(50) not null,
   u_designation varchar(50) not null,
   borrowed_equipment_barcode char(10) not null,
   CONSTRAINT pk_u_user_id PRIMARY KEY(u_user_id),
    )

IF EXISTS(SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE
TABLE_NAME='equipments')
DROP TABLE equipments
go
CREATE TABLE equipments (
   e_barcode char(10) not null,
   e_name varchar(50) not null,
   e_manufacture varchar(50) not null,
   e_model varchar(50) not null,
   e_make varchar(50) not null,
   CONSTRAINT pk_e_barcode PRIMARY KEY(e_barcode),
    )
```

```
IF EXISTS(SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE
TABLE_NAME='activity_loans')
DROP TABLE activity_loans
go
create table activity_loans
 (
 activity_identity int IDENTITY(1,1) NOT NULL,
 activity_equipment_barcode char(10) not null,
 a_url varchar(255) not null, --will contain info about the activity and how to perform it along with
description
 a_poc varchar(50) not null,
 a_lab_no varchar(20) not null,  --will show what is the activity lab number, eg lab1,lab2......
 a_course_name varchar(20) not null, --which course that activity belongs to
 a_borrow_date date not null,
 a_due_date date not null,
 a_reason varchar(50) not null,
  CONSTRAINT pk_activity_loans PRIMARY KEY(activity_identity),
)

IF EXISTS(SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE
TABLE_NAME='adhoc_loans')
DROP TABLE adhoc_loans
go
create table adhoc_loans
 (
 adhoc_identity int IDENTITY(1,1) NOT NULL,
 adhoc_equipment_barcode char(10) not null,
 ad_url varchar(255) not null,
 ad_equipment_name varchar(25),--should match with barcode and equipment name of equipments table
 ad_poc varchar(20) not null,
 ad_email varchar(20) not null,--poc email
 ad_borrow_date date not null,
 ad_due_date date not null,
 ad_reason varchar(20) not null,
   CONSTRAINT pk_adhoc_loans PRIMARY KEY(adhoc_identity),
 )

IF EXISTS(SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE
TABLE_NAME='cent_affiliations')
DROP TABLE cent_affiliations
go
create table cent_affiliations
```

```
 (
 cent_identity int IDENTITY(1,1) NOT NULL,
 c_email varchar(50) not null,
 c_password varchar(50) not null,
 c_firstname varchar(20) not null,
 c_lastname varchar(20) not null,
 c_loaned_items_barcode char(10),
 CONSTRAINT pk_cent_affiliations PRIMARY KEY(cent_identity),
 )

IF EXISTS(SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE
TABLE_NAME='bridge_user_equipment')
DROP TABLE bridge_user_equipment
go
CREATE TABLE bridge_user_equipment (
    bridge_u_identity int not null,
    bridge_e_barcode char(10) not null,
)

-- ADDING CONSTRAINTS TO ALL TABLES

ALTER TABLE adhoc_loans
ADD
CONSTRAINT fk_ad_barcode FOREIGN KEY (adhoc_equipment_barcode) REFERENCES
equipments(e_barcode)


ALTER TABLE activity_loans
ADD
CONSTRAINT fk_activity_barcode FOREIGN KEY (activity_equipment_barcode) REFERENCES
equipments(e_barcode)


ALTER TABLE cent_affiliations
ADD
CONSTRAINT fk_cent_barcode FOREIGN KEY (c_loaned_items_barcode) REFERENCES
equipments(e_barcode)


ALTER TABLE bridge_user_equipment
ADD
CONSTRAINT PK_bridge_user_equipment PRIMARY KEY(bridge_u_identity,bridge_e_barcode),
CONSTRAINT FK_bridge_user
```

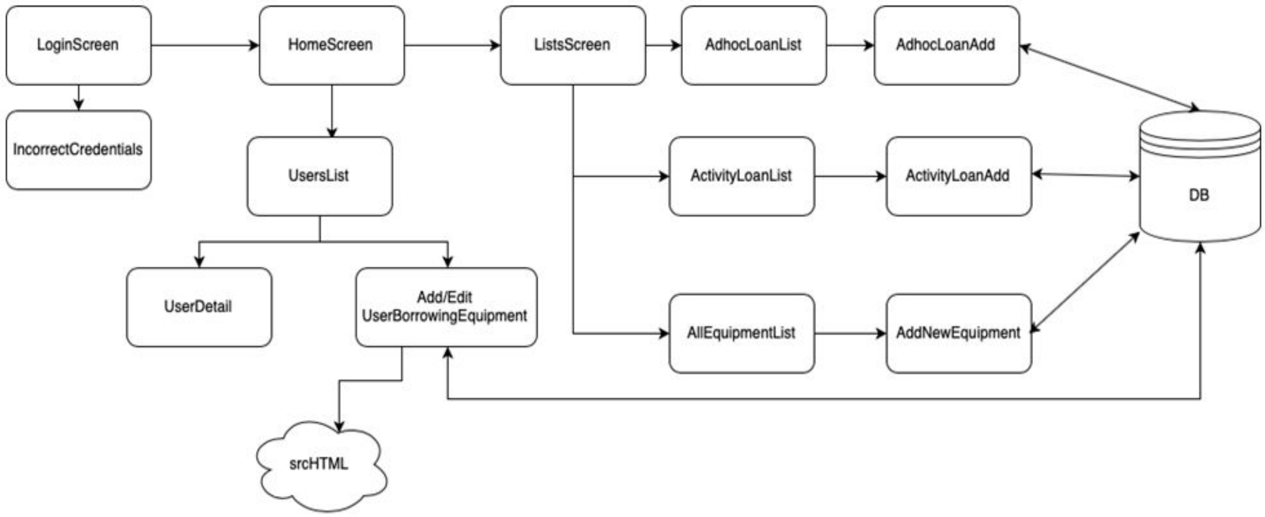
```
    FOREIGN KEY (bridge_u_identity)
    REFERENCES users (u_user_id),
  CONSTRAINT FK_bridge_equipment
    FOREIGN KEY (bridge_e_barcode)
    REFERENCES equipments (e_barcode)


ALTER TABLE users
ADD
CONSTRAINT FK_user_barcode
 FOREIGN KEY (borrowed_equipment_barcode)
 references equipments (e_barcode)


--INSERTING IN EQUIPMENTS
insert into equipments
values(12345,'monitor','dell','OPTIPLEX745','1990')

insert into equipments
values(23456,'pc','dell','OPTIPLEX960','1995')

insert into equipments
values(34567,'monitor','acer','OPTIPLEX790','1996')

insert into equipments
values(45678,'monitor','samsung','OPTIPLEX900','2010')

insert into equipments
values(56789,'pc','lg','OPTIPLEX700','2011')

--INSERTING IN USERS
insert into users
values(1,'shladdha@syr.edu','shladdha','password','shripad','laddha','2019-11-20','2019-12-01','project work','student',12345)

insert into users
values(2,'pdsankpa@syr.edu','pdsankpa','password','prachi','sankpal','2019-10-12','2019-10-19','project work','student',23456)

insert into users
values(3,'aarao@syr.edu','aarao','password','anupama','rao','2019-09-15','2019-09-22','personal','student',23456)
```

```
insert into users
values(4,'mfudge@syr.edu','mfudge','password','michael','fudge','2019-09-27','2019-10-05','research
work','faculty',45678)

insert into users
values(5,'pzhang@syr.edu','pzhang','password','ping','zhang','2019-11-08','2019-11-15','research
work','faculty',56789)


--INSERTING INTO ADHOC LOANS
insert into adhoc_loans
values(12345,'http://dummuadhoc','monitor','poc1','poc1@gmail.com',' 2015-10-19','2015-10-26','my
work')

insert into adhoc_loans
values(12345,'http://dummuadhoc','monitor','poc1','poc1@gmail.com',' 2016-10-19','2016-10-26','my
work1')

insert into adhoc_loans
values(45678,'http://dummuadhoc1','monitor','poc2','poc2@gmail.com',' 2014-10-19','2014-10-26','my
work2')

insert into adhoc_loans
values(23456,'http://dummuadhoc2','pc','poc3','poc3@gmail.com',' 2015-10-18','2014-10-25','my work3')

--INSERTING INTO ACTIVITY LOAN
insert into activity_loans
values(34567,'http://dummuactivity','poc11@gmail.com','lab1','ist659',' 2010-10-18','2010-10-25','lab
work')

insert into activity_loans
values(34567,'http://dummuactivity','poc11@gmail.com','lab1','ist659',' 2011-12-08','2011-12-15','lab
work')

insert into activity_loans
values(56789,'http://dummuactivity1','poc12@gmail.com','lab2','ist659',' 2013-10-23','2010-10-30','lab
work1')

--INSERTING VALUES IN CENT TABLE
insert into cent_affiliations
values('avdeshmukh@syr.edu','password','avin','deskhmukh',12345)

insert into cent_affiliations
```

```
values('lptalreja@syr.edu','password','lavnish','talreja',23456)

insert into cent_affiliations
values('avdeshmukh@syr.edu','password','avin','deskhmukh',12345)

insert into cent_affiliations
values('avdeshmukh@syr.edu','password','avin','deskhmukh',23456)

--INSERTING INTO BRIDGE TABLE
insert into bridge_user_equipment
values(2,12345)

insert into bridge_user_equipment
values(1,23456)

insert into bridge_user_equipment
values(1,34567)

insert into bridge_user_equipment
values(3,23456)

insert into bridge_user_equipment
values(4,45678)


select * from users
select * from equipments
select * from adhoc_loans
select * from activity_loans
select * from cent_affiliations
select * from bridge_user_equipment

--DROPPING AND CREATING VIEWS
if exists(SELECT * FROM INFORMATION_SCHEMA.VIEWS WHERE
TABLE_NAME='v_unique_adhoc_loans')
drop view v_unique_adhoc_loans
go
create view v_unique_adhoc_loans as
 select * from adhoc_loans where adhoc_equipment_barcode in (select distinct
adhoc_equipment_barcode from adhoc_loans)
 go

--DROPPING AND CREATING VIEWS
```

```
if exists(SELECT * FROM INFORMATION_SCHEMA.VIEWS WHERE
TABLE_NAME='v_unique_activity_loans')
drop view v_unique_activity_loans
go
create view v_unique_activity_loans as
 select * from activity_loans where activity_equipment_barcode in
 (select distinct activity_equipment_barcode from activity_loans)
 go
```

5. Diagram of Screens used in the application

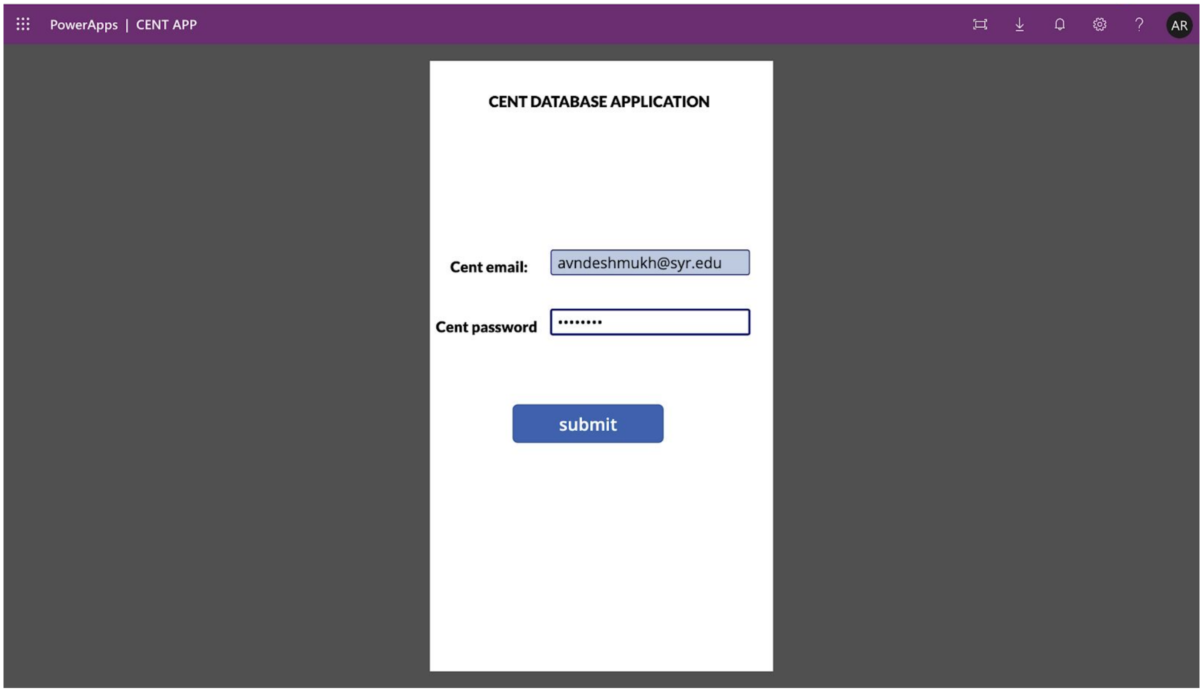


6. Implementation of the application in Power Apps

- **AppLink**:
  https://apps.powerapps.com/play/d98a27f2-08b1-44a0-b4e1-26b0eaea7d03?tenantId=4278a402-1a9e-4eb9-8414-ffb55a5fcf1e

- **Login Screen**

  If cent affiliated person exists in cent affiliated table in the database then only he/she will be able to login otherwise there will be an error popup

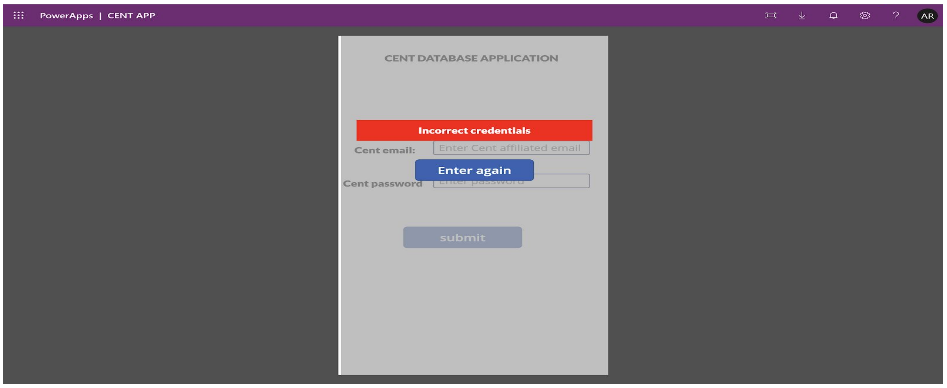


Cent email: avndeshmukh@syr.edu

Cent password ••••••••

- **Incorrect credentials**

  If the user enters incorrect credentials, he/she gets this error popup with an Enter again button which takes the user back to the login screen



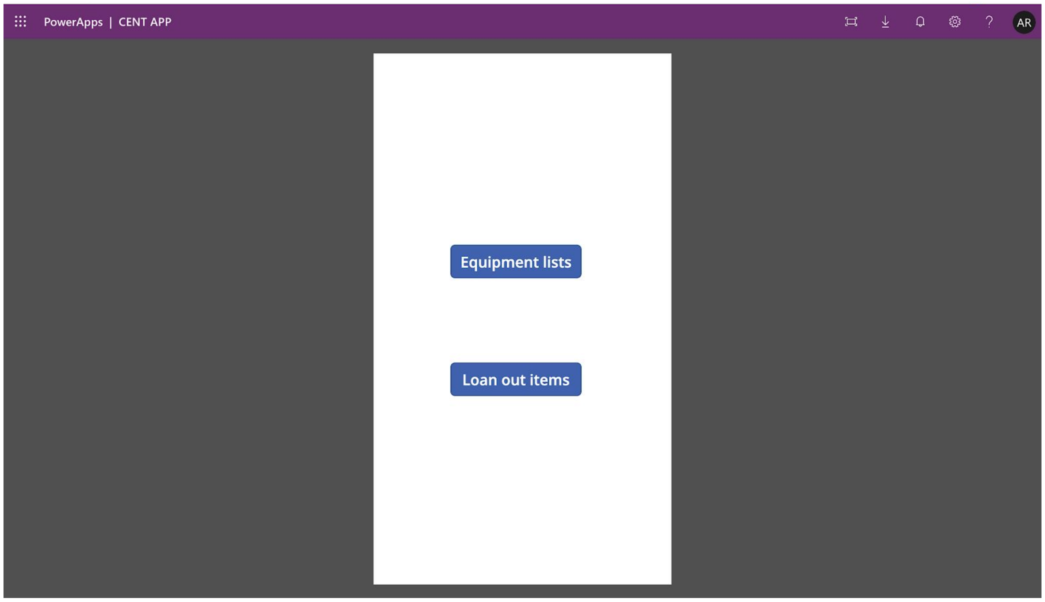
Incorrect credentials

Cent email: Enter Cent affiliated email

Cent password Enter password

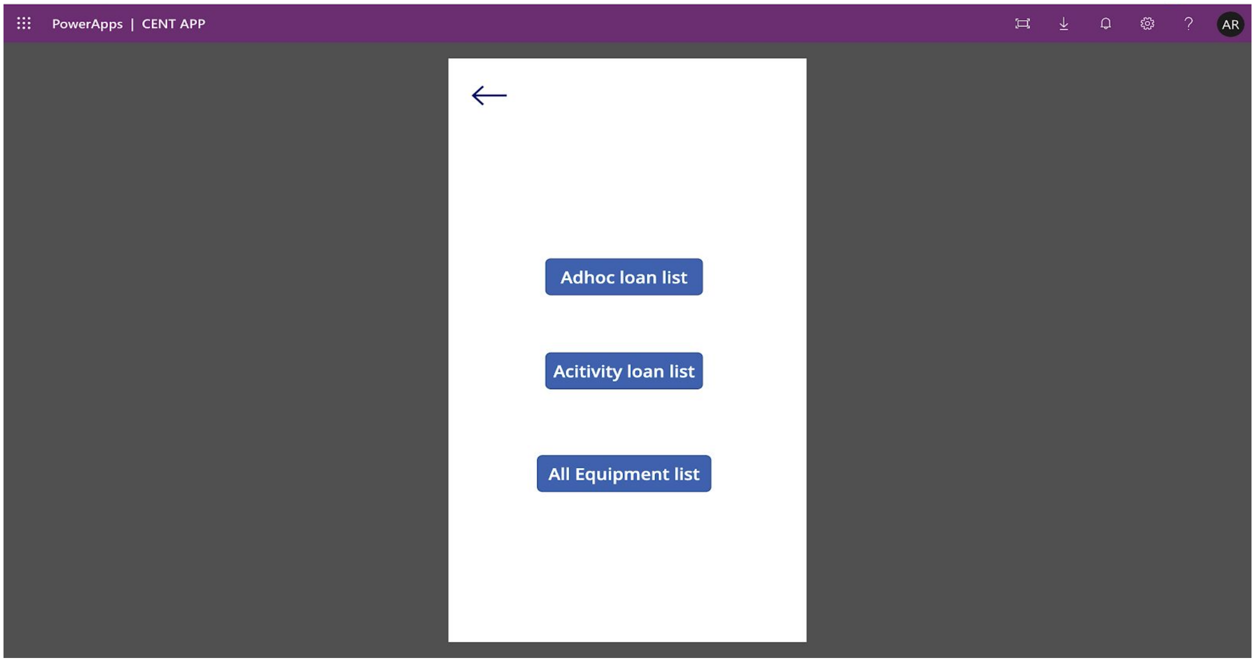
- Home Screen

After entering the correct credentials, the user is able to see two buttons through which he/she can loan out a new item(if he/she click on "Loan out items" button) in the next screen or view Adhoc loan list /Activity loan list/ Equipments list by clicking on the "Equipment list" button



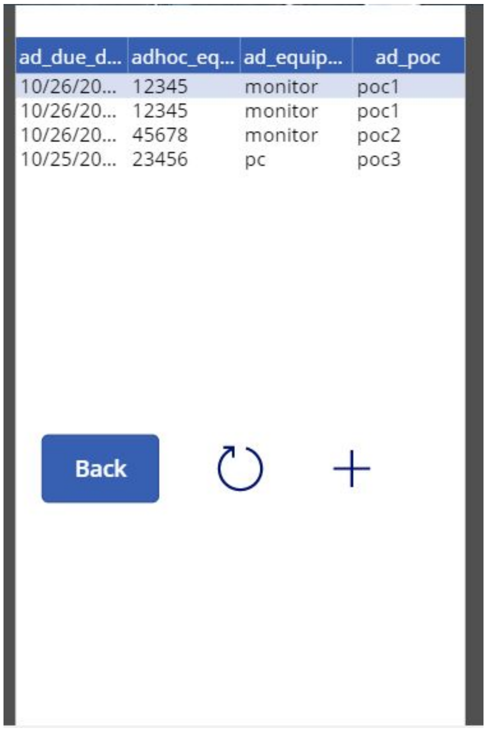
- **Lists Screen**

When the user clicks on "Equipment list" button, it gets directed to lists screen displaying three buttons of "Adhoc Loan list", "Activity loan list", "All Equipment list" so that it can choose the option.



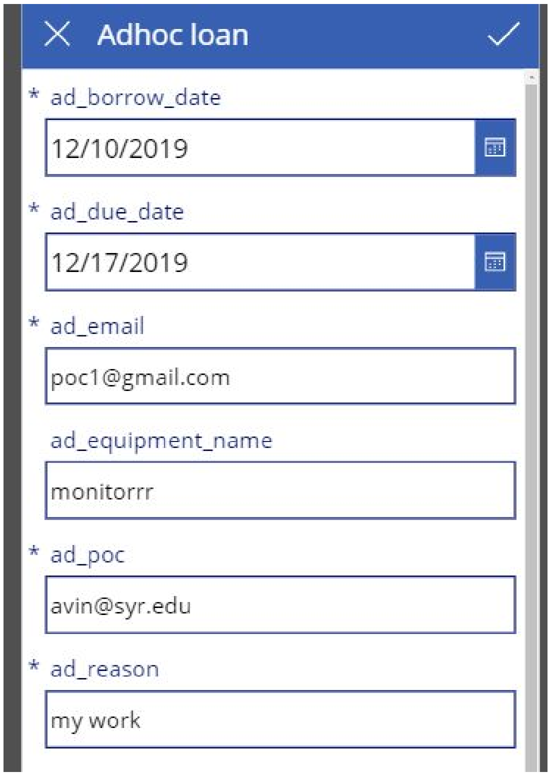
- **Adhoc Loan List**

  This list displays the list of all adhoc loans from the adhoc_loans table, user can add an adhoc loan by clicking on "+"



ad_due_d...	adhoc_eq...	ad_equip...	ad_poc
10/26/20...	12345	monitor	poc1
10/26/20...	12345	monitor	poc1
10/26/20...	45678	monitor	poc2
10/25/20...	23456	pc	poc3

- **The user enters the details for adhoc loan**



Adhoc loan

* ad_borrow_date

12/10/2019

* ad_due_date

12/17/2019

* ad_email

poc1@gmail.com

ad_equipment_name

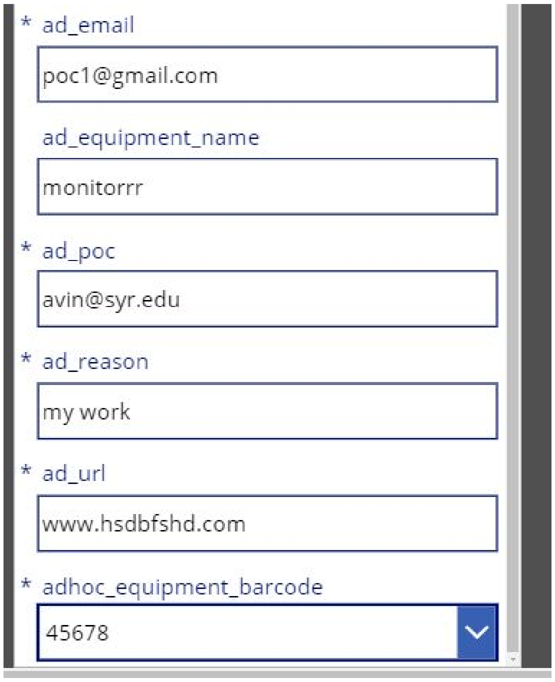
monitorrr

* ad_poc

avin@syr.edu

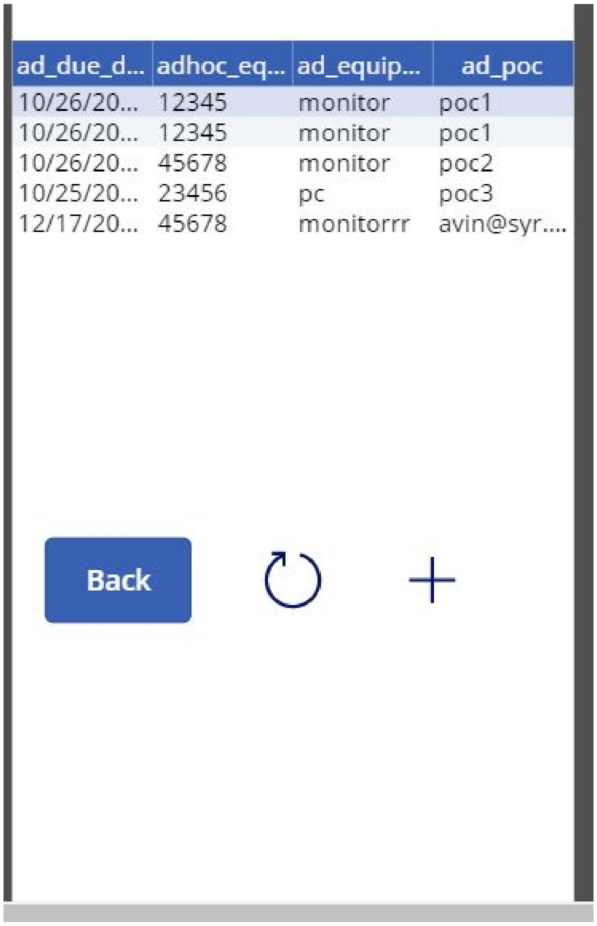
* ad_reason

my work

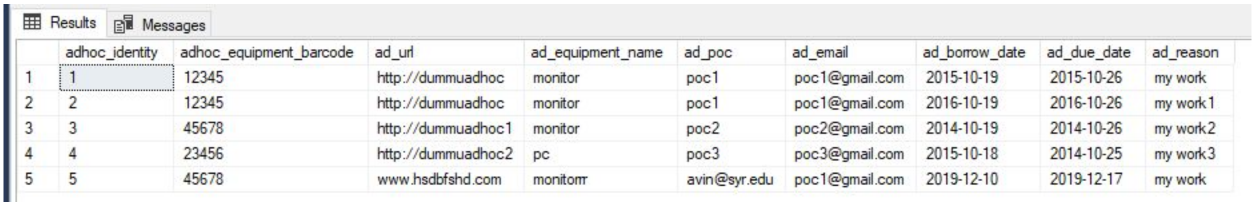


* ad_email	poc1@gmail.com
ad_equipment_name	monitorrr
* ad_poc	avin@syr.edu
* ad_reason	my work
* ad_url	www.hsdbfshd.com
* adhoc_equipment_barcode	45678

- **When the user hits the “✓” the adhoc loan is created and stored in adhoc_loans table and instantly gets updated in the previous screen.**



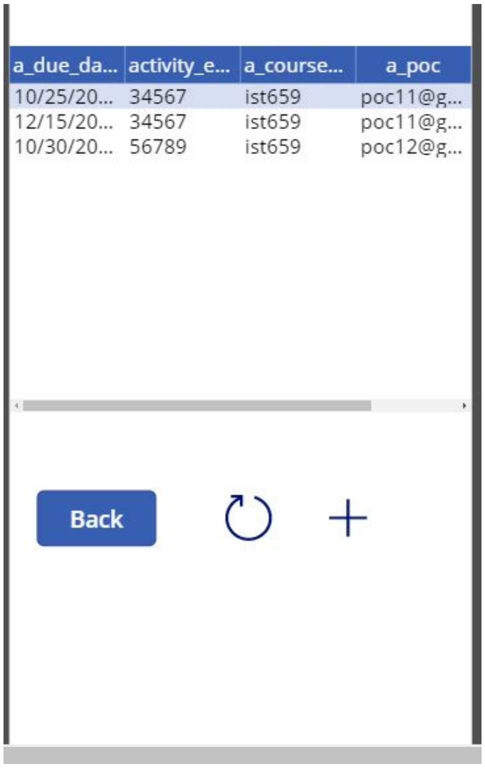
ad_due_d...	adhoc_eq...	ad_equip...	ad_poc
10/26/20...	12345	monitor	poc1
10/26/20...	12345	monitor	poc1
10/26/20...	45678	monitor	poc2
10/25/20...	23456	pc	poc3
12/17/20...	45678	monitorrr	avin@syr....



	adhoc_identity	adhoc_equipment_barcode	ad_url	ad_equipment_name	ad_poc	ad_email	ad_borrow_date	ad_due_date	ad_reason
1	1	12345	http://dummuadhoc	monitor	poc1	poc1@gmail.com	2015-10-19	2015-10-26	my work
2	2	12345	http://dummuadhoc	monitor	poc1	poc1@gmail.com	2016-10-19	2016-10-26	my work1
3	3	45678	http://dummuadhoc1	monitor	poc2	poc2@gmail.com	2014-10-19	2014-10-26	my work2
4	4	23456	http://dummuadhoc2	pc	poc3	poc3@gmail.com	2015-10-18	2014-10-25	my work3
5	5	45678	www.hsdbfshd.com	monitorr	avin@syr.edu	poc1@gmail.com	2019-12-10	2019-12-17	my work

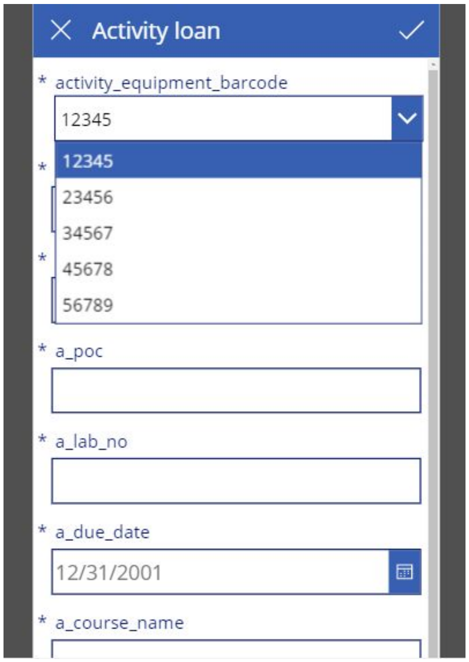
- **Activity loan**

  This list displays the list of all activityloans from the activity_loans table, user can add an adhoc loan by clicking on "+". Here the assumption is that, the activity has a barcode on the activity bundle and the a_url will have all the details about that activity including all the equipments in the bundle and how to use them.



a_due_da...	activity_e...	a_course...	a_poc
10/25/20...	34567	ist659	poc11@g...
12/15/20...	34567	ist659	poc11@g...
10/30/20...	56789	ist659	poc12@g...

The user enters the details for activity loan



Activity loan

* activity_equipment_barcode

12345

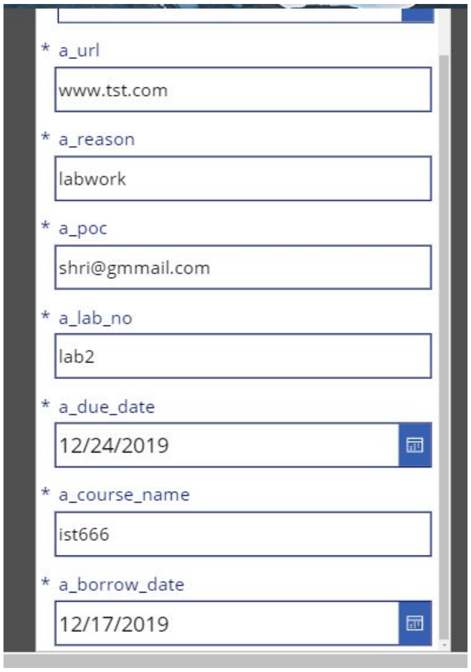
* a_poc

* a_lab_no

* a_due_date

12/31/2001

* a_course_name



* a_url

www.tst.com

* a_reason

labwork

* a_poc

shri@gmmail.com

* a_lab_no

lab2

* a_due_date

12/24/2019

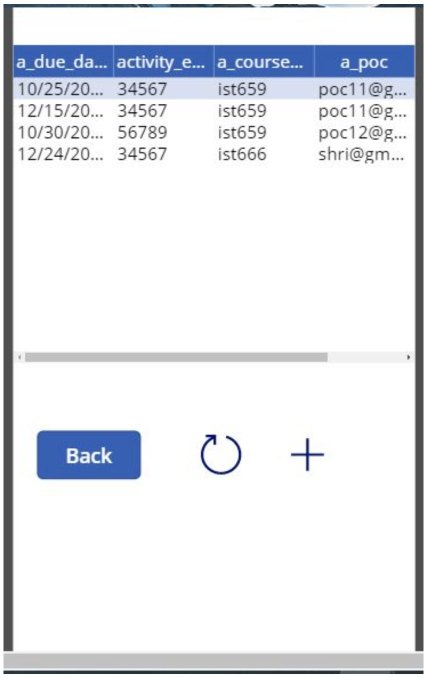
* a_course_name

ist666

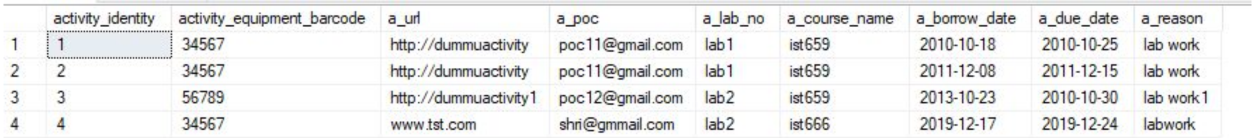
* a_borrow_date

12/17/2019

The changes get reflected in the list above along with entry in database



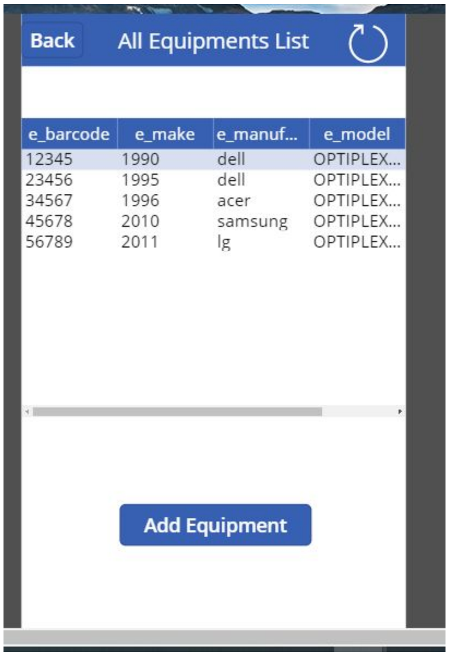
a_due_da...	activity_e...	a_course...	a_poc
10/25/20...	34567	ist659	poc11@g...
12/15/20...	34567	ist659	poc11@g...
10/30/20...	56789	ist659	poc12@g...
12/24/20...	34567	ist666	shri@gm...



	activity_identity	activity_equipment_barcode	a_url	a_poc	a_lab_no	a_course_name	a_borrow_date	a_due_date	a_reason
1	1	34567	http://dummuactivity	poc11@gmail.com	lab1	ist659	2010-10-18	2010-10-25	lab work
2	2	34567	http://dummuactivity	poc11@gmail.com	lab1	ist659	2011-12-08	2011-12-15	lab work
3	3	56789	http://dummuactivity1	poc12@gmail.com	lab2	ist659	2013-10-23	2010-10-30	lab work1
4	4	34567	www.tst.com	shri@gmmail.com	lab2	ist666	2019-12-17	2019-12-24	labwork

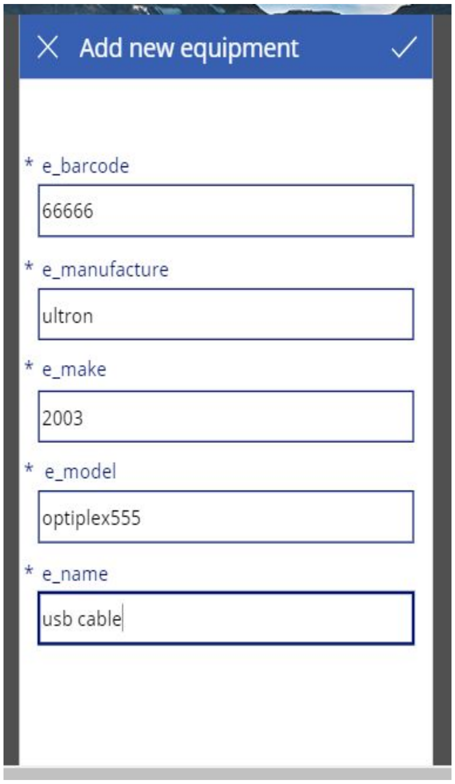
All equipments list

Shows the list of all equipments that can be loaned out. Assuming all equipmemts are available



e_barcode	e_make	e_manuf...	e_model
12345	1990	dell	OPTIPLEX...
23456	1995	dell	OPTIPLEX...
34567	1996	acer	OPTIPLEX...
45678	2010	samsung	OPTIPLEX...
56789	2011	lg	OPTIPLEX...

Add a new equipment



* e_barcode: 66666

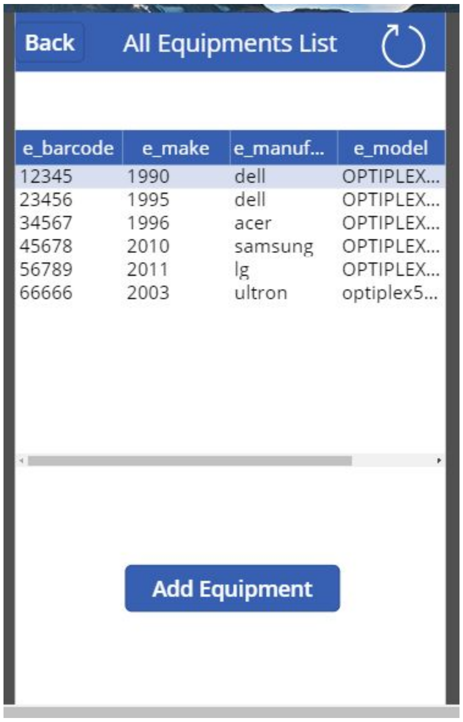
* e_manufacture: ultron

* e_make: 2003

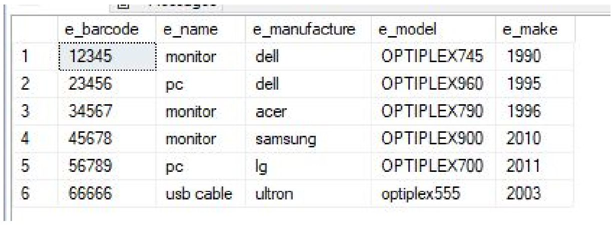
* e_model: optiplex555

* e_name: usb cable

The equipment get added in list as well as database when you click on “✓ ”



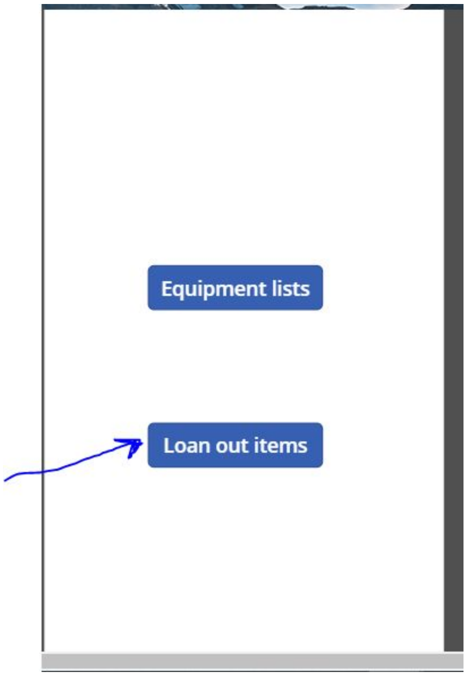
e_barcode	e_make	e_manuf...	e_model
12345	1990	dell	OPTIPLEX...
23456	1995	dell	OPTIPLEX...
34567	1996	acer	OPTIPLEX...
45678	2010	samsung	OPTIPLEX...
56789	2011	lg	OPTIPLEX...
66666	2003	ultron	optiplex5...



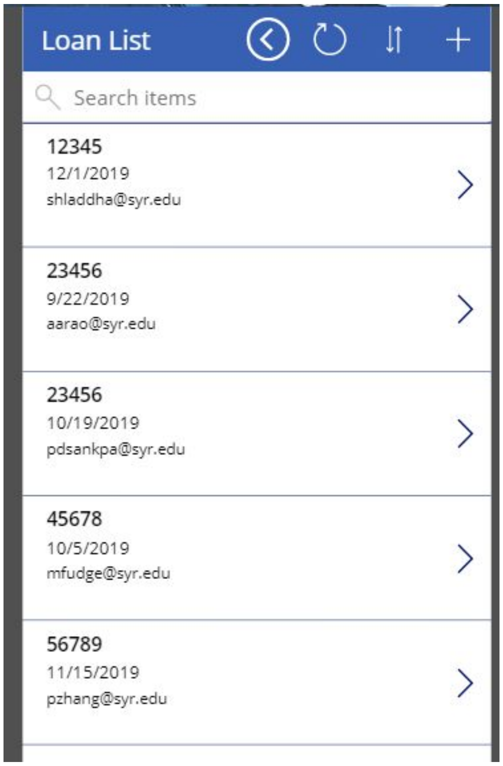
	e_barcode	e_name	e_manufacture	e_model	e_make
1	12345	monitor	dell	OPTIPLEX745	1990
2	23456	pc	dell	OPTIPLEX960	1995
3	34567	monitor	acer	OPTIPLEX790	1996
4	45678	monitor	samsung	OPTIPLEX900	2010
5	56789	pc	lg	OPTIPLEX700	2011
6	66666	usb cable	ultron	optiplex555	2003

Loan list

When you click on the loan out items button



User can see all the list of equipments that have been loaned out along with the person who took it and the due date



12345
12/1/2019
shladdha@syr.edu

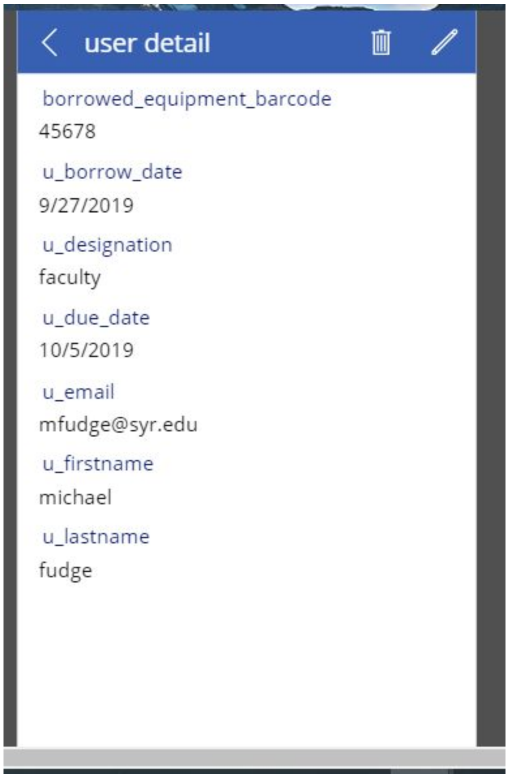
23456
9/22/2019
aarao@syr.edu

23456
10/19/2019
pdsankpa@syr.edu

45678
10/5/2019
mfudge@syr.edu

56789
11/15/2019
pzhang@syr.edu

Borrower details



borrowed_equipment_barcode
45678

u_borrow_date
9/27/2019

u_designation
faculty

u_due_date
10/5/2019

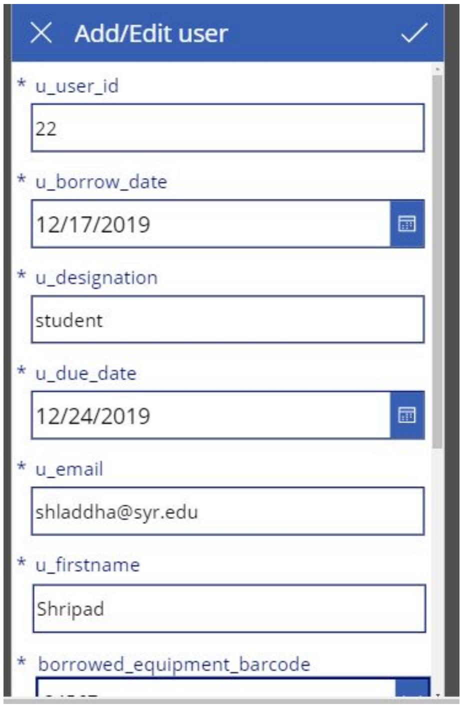
u_email
mfudge@syr.edu

u_firstname
michael

u_lastname
fudge

- **Add/edit user borrowing equipment**

When you click on the ✓ button, the entry gets added to user table and the list is updated.



* u_user_id
22

* u_borrow_date
12/17/2019

* u_designation
student

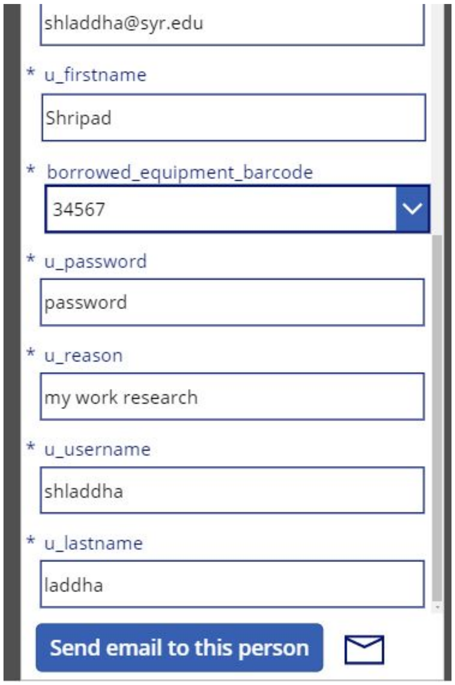
* u_due_date
12/24/2019

* u_email
shladdha@syr.edu

* u_firstname
Shripad

* borrowed_equipment_barcode

Entering details to create a new borrower request, when you click the "send email to this person" buttonit sends the borrow date and due date to the persons details gathered from here and storing in an html screen.


shladdha@syr.edu

* u_firstname
Shripad

* borrowed_equipment_barcode
34567

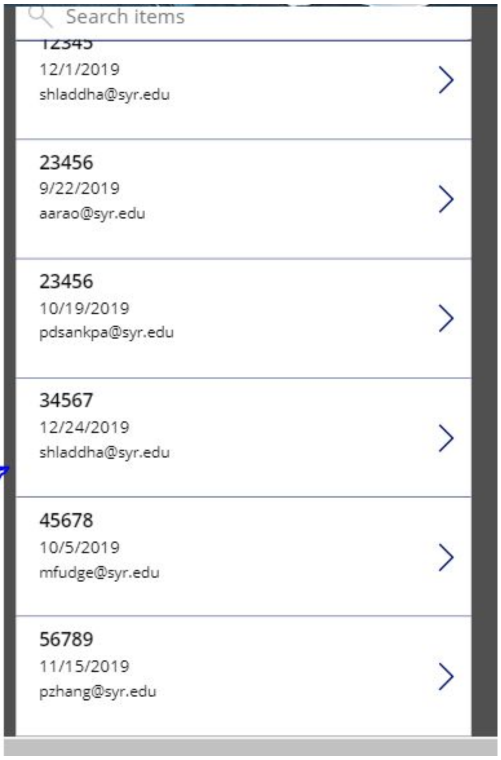
* u_password
password

* u_reason
my work research

* u_username
shladdha

* u_lastname
laddha

User can search by email/barcode/due date and get to know who has which equipment and till when



12345
12/1/2019
shladdha@syr.edu

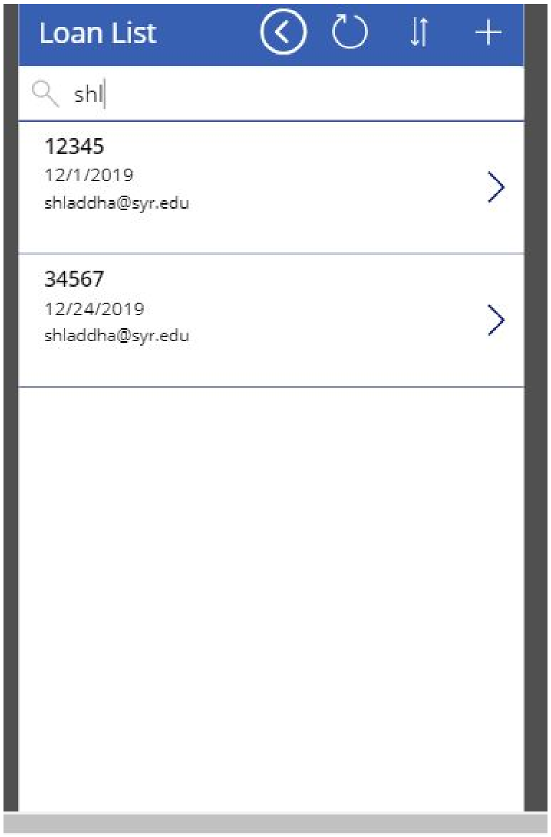
23456
9/22/2019
aarao@syr.edu

23456
10/19/2019
pdsankpa@syr.edu

34567
12/24/2019
shladdha@syr.edu

45678
10/5/2019
mfudge@syr.edu

56789
11/15/2019
pzhang@syr.edu



shl

12345
12/1/2019
shladdha@syr.edu

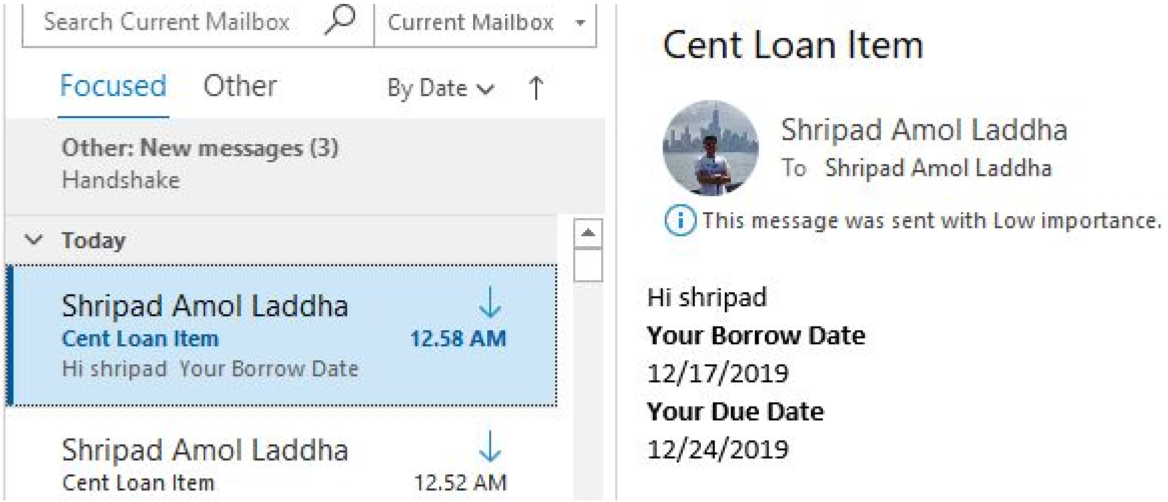
34567
12/24/2019
shladdha@syr.edu

Email receipt:

Used this on the “send email button”- Office365Outlook.SendEmailV2(DataCardValue14,"Cent Loan Item",Emailhtml.HtmlText)

The Html.Text text file contains Html text:

"<div>Hi "& DataCardValue15 & "</div><div style='font-weight:bold'>Your Borrow Date</div><div>"& DataCardValue11 & "</div><div style='font-weight:bold'>Your Due Date</div><div>"& DataCardValue13 & "</div>" **which fetches information to be sent out on email dynamically.**



Shripad Amol Laddha
Cent Loan Item 12.58 AM
Hi shripad Your Borrow Date

Shripad Amol Laddha
Cent Loan Item 12.52 AM

Cent Loan Item

Shripad Amol Laddha
To Shripad Amol Laddha

This message was sent with Low importance.

Hi shripad

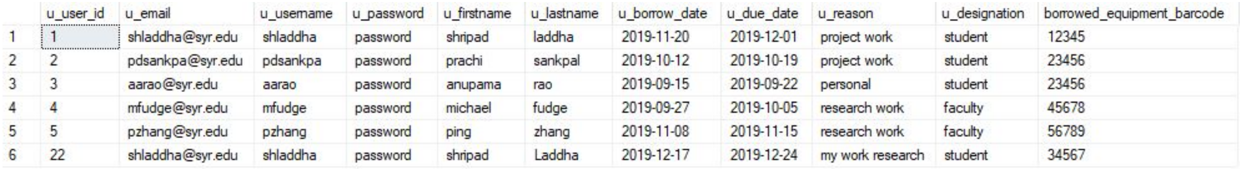
Your Borrow Date

12/17/2019

Your Due Date

12/24/2019

The inserted value can be seen in the database with due date.



	u_user_id	u_email	u_usemame	u_password	u_firstname	u_lastname	u_borrow_date	u_due_date	u_reason	u_designation	borrowed_equipment_barcode
1	1	shladdha@syr.edu	shladdha	password	shripad	laddha	2019-11-20	2019-12-01	project work	student	12345
2	2	pdsankpa@syr.edu	pdsankpa	password	prachi	sankpal	2019-10-12	2019-10-19	project work	student	23456
3	3	aarao@syr.edu	aarao	password	anupama	rao	2019-09-15	2019-09-22	personal	student	23456
4	4	mfudge@syr.edu	mfudge	password	michael	fudge	2019-09-27	2019-10-05	research work	faculty	45678
5	5	pzhang@syr.edu	pzhang	password	ping	zhang	2019-11-08	2019-11-15	research work	faculty	56789
6	22	shladdha@syr.edu	shladdha	password	shripad	Laddha	2019-12-17	2019-12-24	my work research	student	34567

7.Team contribution log report

When we initially wanted to start working on the project, we did individual analysis of the requirements and came up with a few different approaches. Due to clashing schedules, we divided up the work and started working on it individually.

Shripad started off the project by doing analysis for conceptual and logical modelling and jotting down the attributes, entities and relationships on paper.

Anupama was responsible for converting the diagrams on paper into the actual data flow diagram that is conceptual and logical model for the database.

Prachi converted the conceptual & logical models into tables on the SQL server. With the initial phase of the project complete, we decided it would be best to start working on the rest of the project together because it needed a lot of team-work contribution. Over the Thanksgiving break, we met 2 times and spent at least 9 hours building the project ground-up. Overall, we were able to implement most of the major functionalities.

CENT DATABASE MANAGEMENT SYSTEM

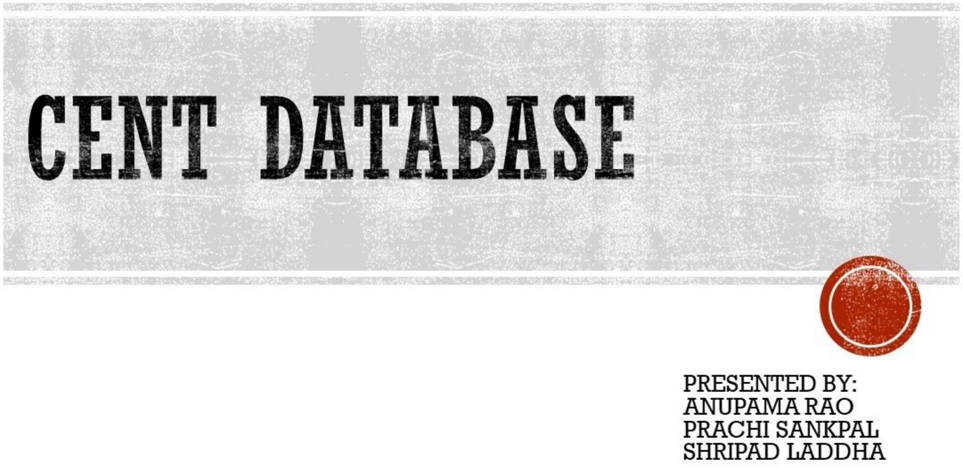
DATE	ITEM	NOTES
11-10-2019	GROUP DISCUSSION	1. Review of CENT requirements
		2. Decided the flow of project
11/24/19	INDIVIDUAL WORK	Anupama- Conceptual and Logical modelling
		Shripad - Conceptual and Logical modelling
		Prachi - Conceptual and Logical modelling , Creation of tables in MSSQL
11/25/19	INDIVIDUAL WORK/GROUP	Anupama- Conceptual and Logical modelling corrections

	DISCUSSION	
		Shripad - Insertion of data into database
		Prachi - Changing tables as per new modeling
12-01-2019	GROUP DISCUSSION	Fixing data flow, whereever necessary, had a few constraints on inserting data
	INDIVIDUAL WORK	Anupama - Designed UI for PowerApps user side
		Shripad - Designed UI for PowerApps client side
		Prachi - Adding functionalities to PowerApps as per need, and fixing incomplete functionalities
12-02-2019	INDIVIDUAL WORK	Anupama - Report, editing videos and presentation
		Shripad - Added email functionality, refined UI, worked with the report and PowerApps demo
		Prachi - Refined UI, PPT and presentation

8.Team presentation, slides and demo

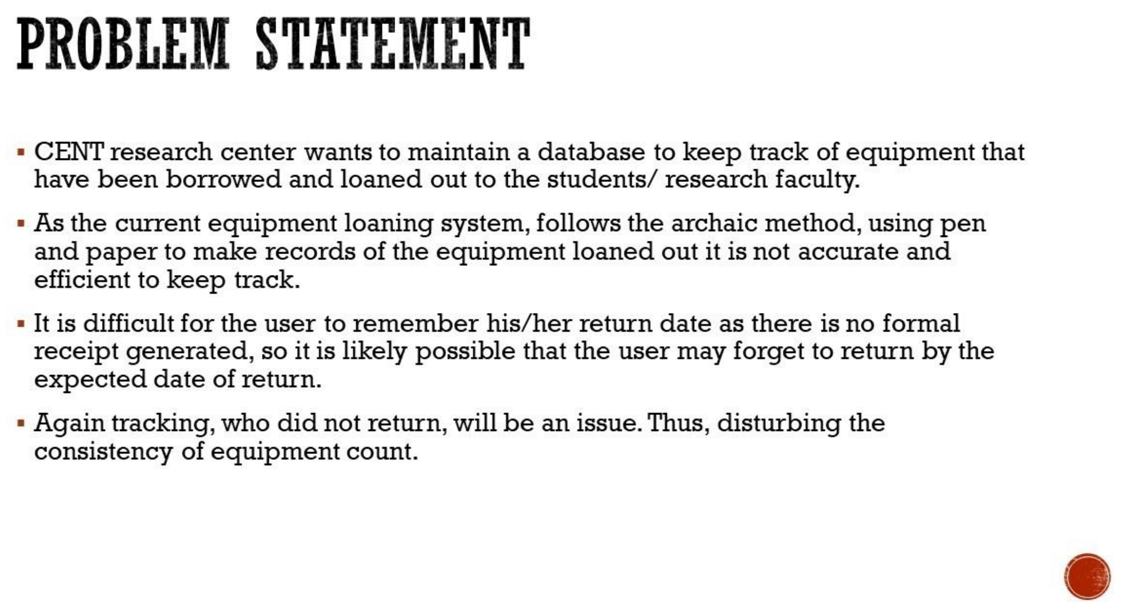
The link demo is given below
Project Demo: https://www.youtube.com/watch?v=Lf-rkn6AAY0&feature=youtu.be

The following represents our presentation slides along with explanations in the youtube link
https://www.youtube.com/watch?v=7GERtb6I3X4&feature=youtu.be



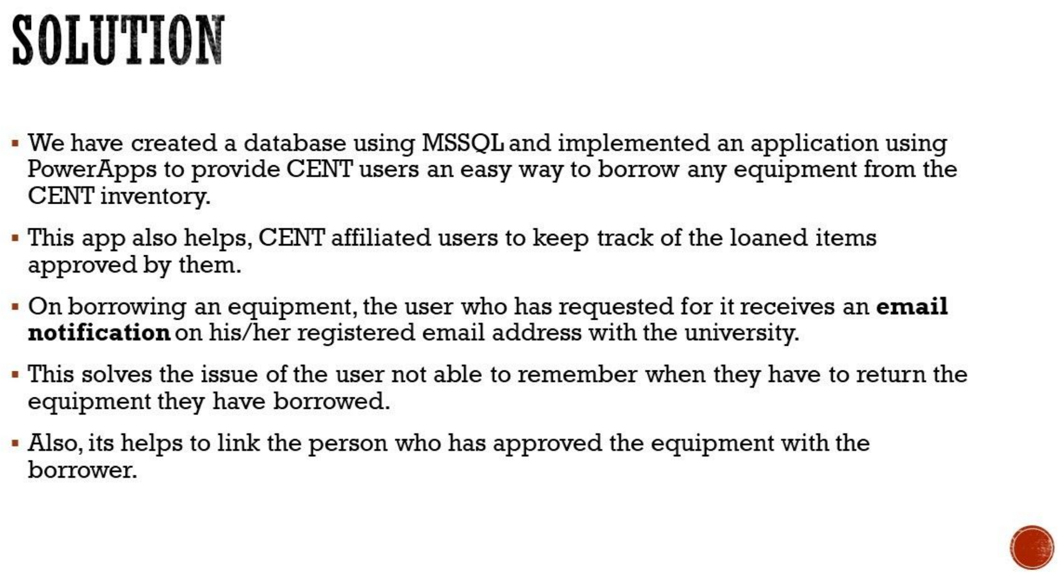
CENT DATABASE

PRESENTED BY:
ANUPAMA RAO
PRACHI SANKPAL
SHRIPAD LADDHA



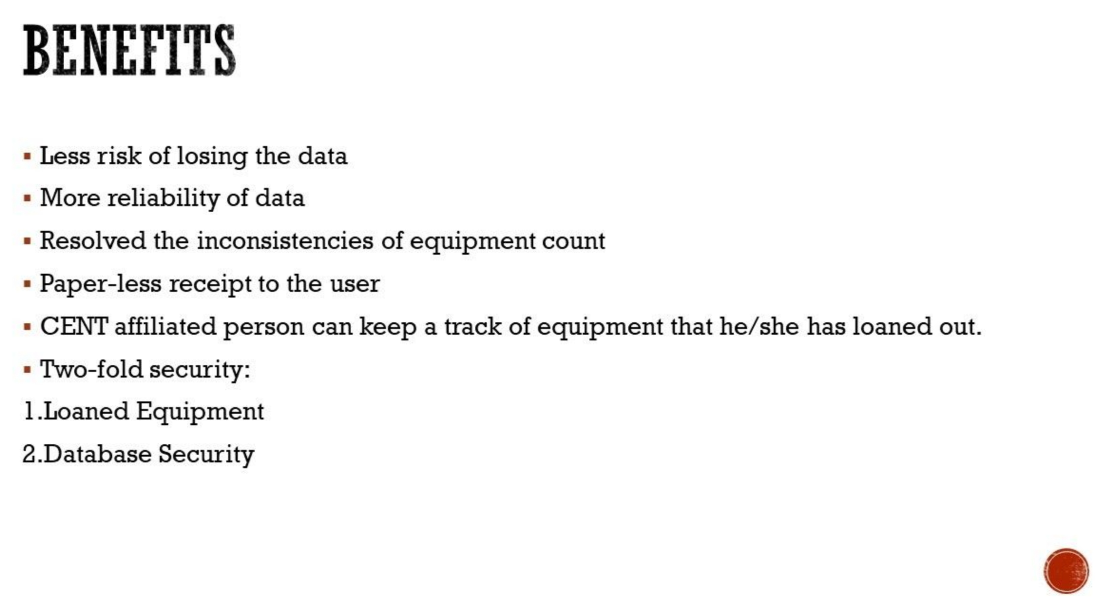
PROBLEM STATEMENT

- CENT research center wants to maintain a database to keep track of equipment that have been borrowed and loaned out to the students/ research faculty.
- As the current equipment loaning system, follows the archaic method, using pen and paper to make records of the equipment loaned out it is not accurate and efficient to keep track.
- It is difficult for the user to remember his/her return date as there is no formal receipt generated, so it is likely possible that the user may forget to return by the expected date of return.
- Again tracking, who did not return, will be an issue. Thus, disturbing the consistency of equipment count.



SOLUTION

- We have created a database using MSSQL and implemented an application using PowerApps to provide CENT users an easy way to borrow any equipment from the CENT inventory.
- This app also helps, CENT affiliated users to keep track of the loaned items approved by them.
- On borrowing an equipment, the user who has requested for it receives an **email notification** on his/her registered email address with the university.
- This solves the issue of the user not able to remember when they have to return the equipment they have borrowed.
- Also, its helps to link the person who has approved the equipment with the borrower.

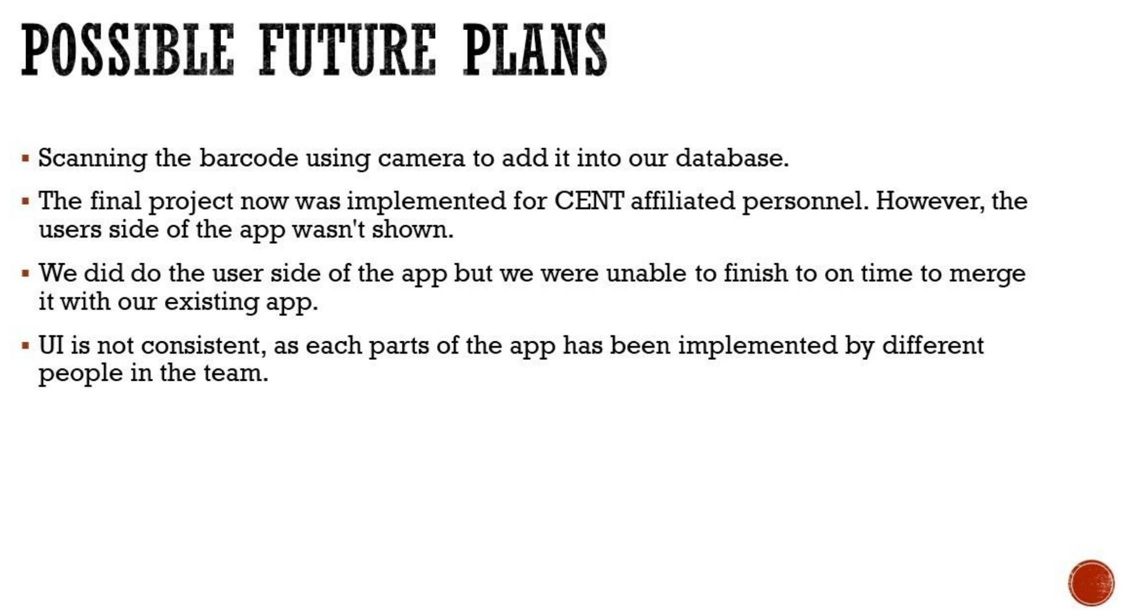


BENEFITS

- Less risk of losing the data
- More reliability of data
- Resolved the inconsistencies of equipment count
- Paper-less receipt to the user
- CENT affiliated person can keep a track of equipment that he/she has loaned out.
- Two-fold security:

1.Loaned Equipment

2.Database Security



POSSIBLE FUTURE PLANS

- Scanning the barcode using camera to add it into our database.
- The final project now was implemented for CENT affiliated personnel. However, the users side of the app wasn't shown.
- We did do the user side of the app but we were unable to finish to on time to merge it with our existing app.
- UI is not consistent, as each parts of the app has been implemented by different people in the team.

9. Video reflection

The link to video reflections which includes the key things our team learned, common pitfalls that we encountered and what we would have done with more time is: https://www.youtube.com/watch?v=_GBW03CQzlo&feature=youtu.be