

UNG Parking Services System



JORDAN TODD

STEPHANIE VERCELLOTTI

BRITTANY WARD

Planning & Selection

1 / 2
ACTIVITIES

1. Identification of Need

- What problems have we encountered?
- What could be improved?

UNG outsources current parking system

- | | |
|-------------------|-------------------|
| • Featureless | • Poor Interface |
| • Limited Support | • Limited Control |

Planning & Selection

2 / 2
ACTIVITIES

2. Investigation & Determination of Scope

- Is it feasible?
 - Can we do it?
- Are the benefits worth it?

Economic Feasibility

- Tangible Benefits
 - Cost Reduction
 - Error Reduction
 - Increased Control
- Intangible Benefits
 - Increased Reputation

Technical Feasibility

Current parking website doesn't seem complex

Schedule Feasibility

- Time-frame can be met
- For at least meeting functional requirements

Scope

Functional Requirements			
Secure Access	Info Updates	View Citations	Check Balances
Paying Fees	Register Vehicles	Request Permits	Manage Account

Implementation Plan		
Component	Function	Source
<i>Database</i>	Collects, stores, retrieves & manipulates user data for use in the web app	Developed In-House using SQL Server
<i>App Server</i>	Processes data requests for transmission between users and the database	Developed In-House using C# & the .NET Framework
<i>Web App</i>	Provides a web portal for users to view account data & for admin to maintain the features and services offered	Developed In-House using the .NET Framework

Analysis: *Determining Requirements*

Study Current Systems & Procedures

Determine
Requirements



Id. Potential
Solutions



Choose Best
Solution

Requirement Collection Method

*Traditional
Business
Document
Analysis*

Discover issues, rules, & directions

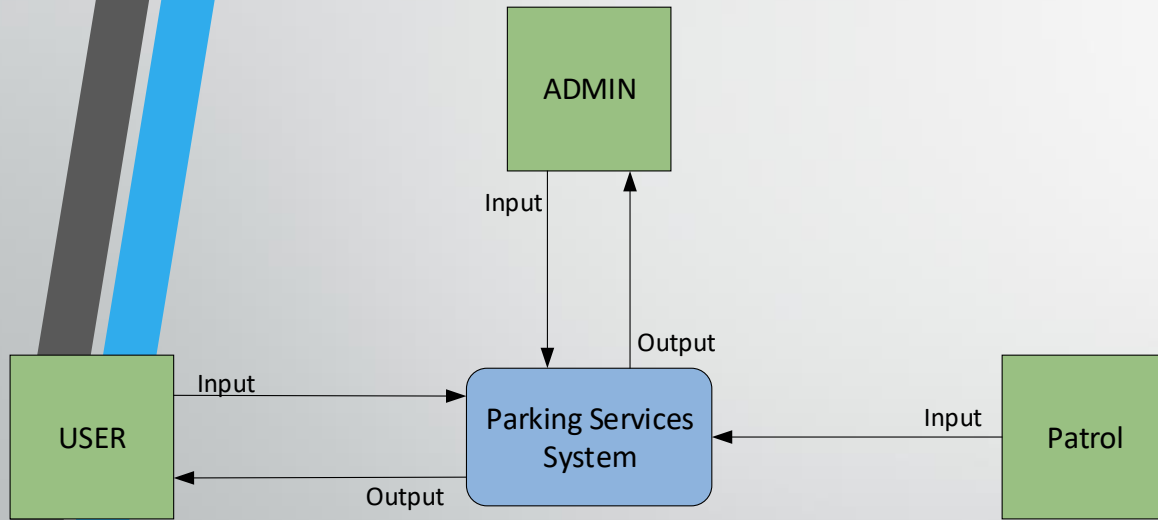
See concrete examples of data & information use



Analysis: *Structuring Requirements* – PROCESS MODELING

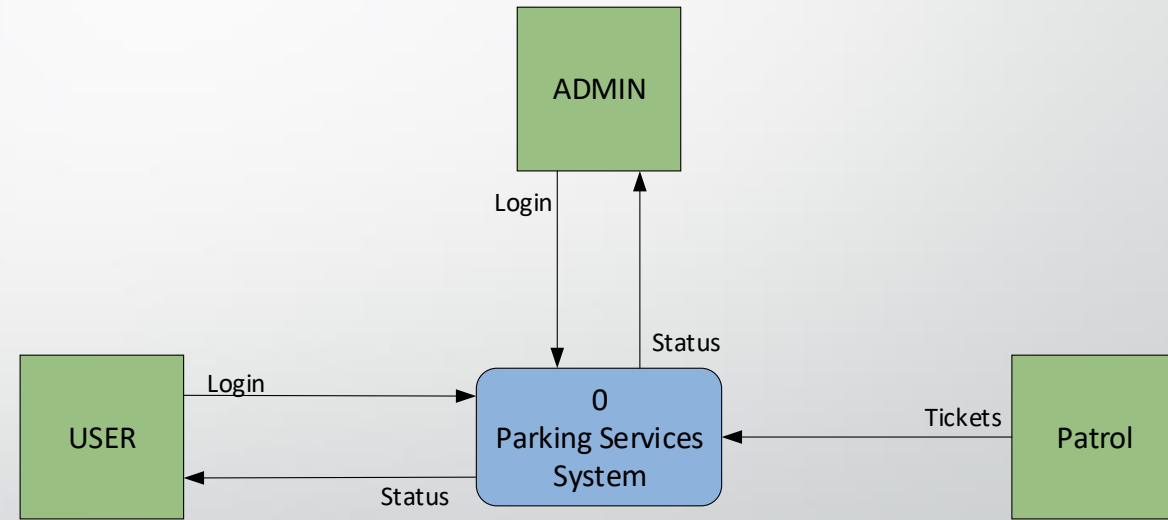
Parking Services System

Context Level DFD



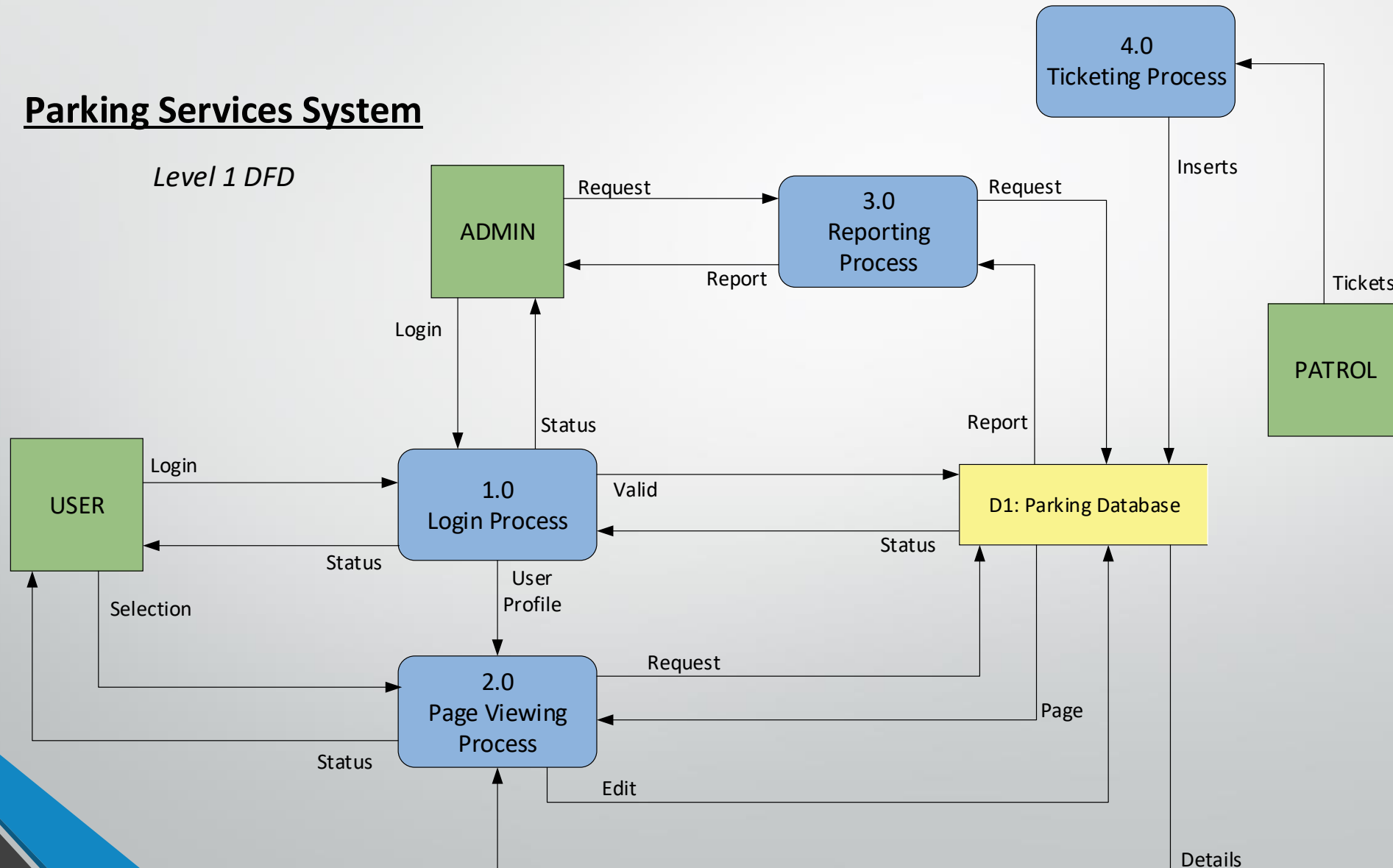
Parking Services System

0 – Level DFD



Analysis: PROCESS MODELING CONT'D

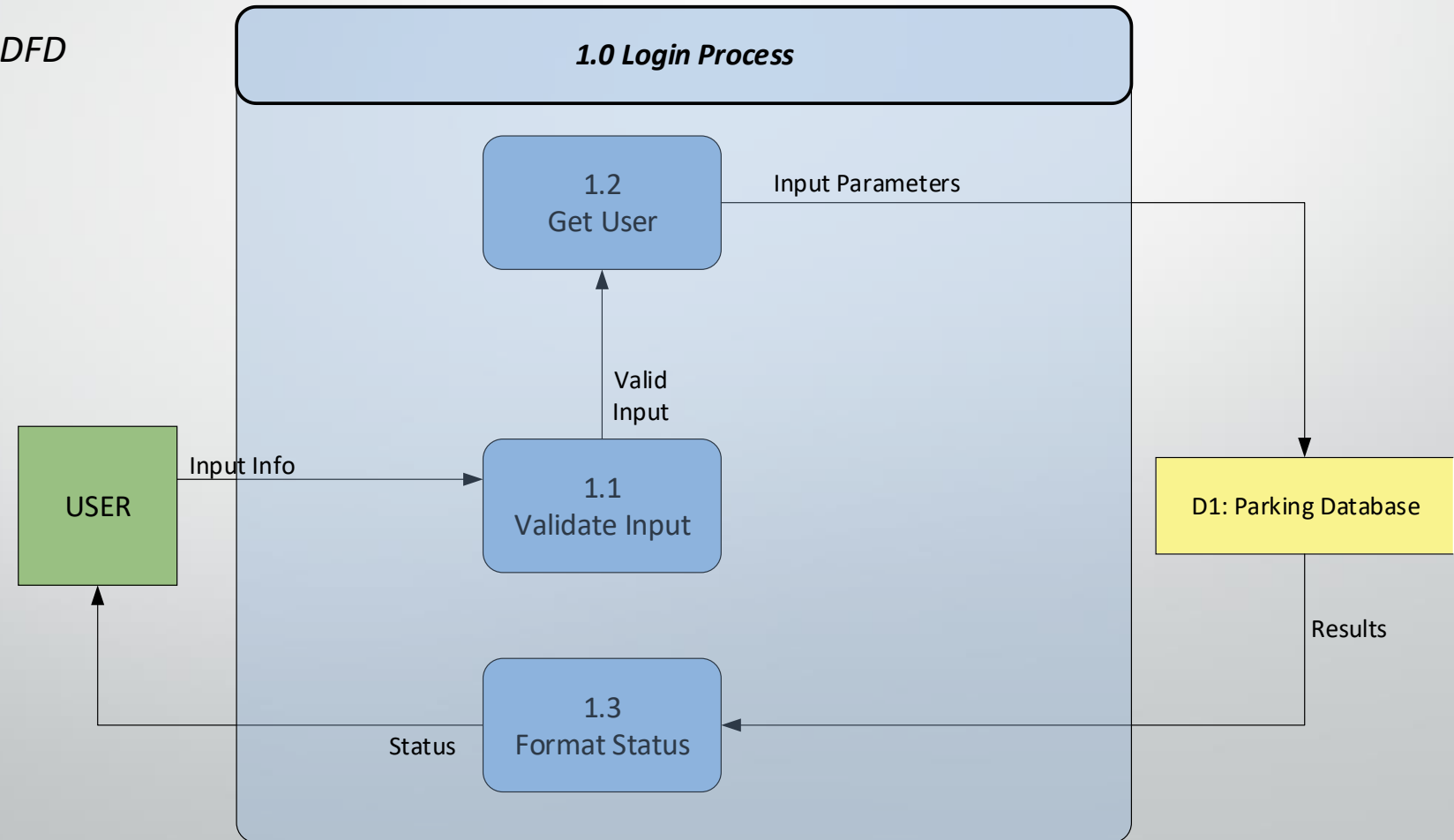
Parking Services System



Analysis: PROCESS MODELING CONT'D

Parking Services System

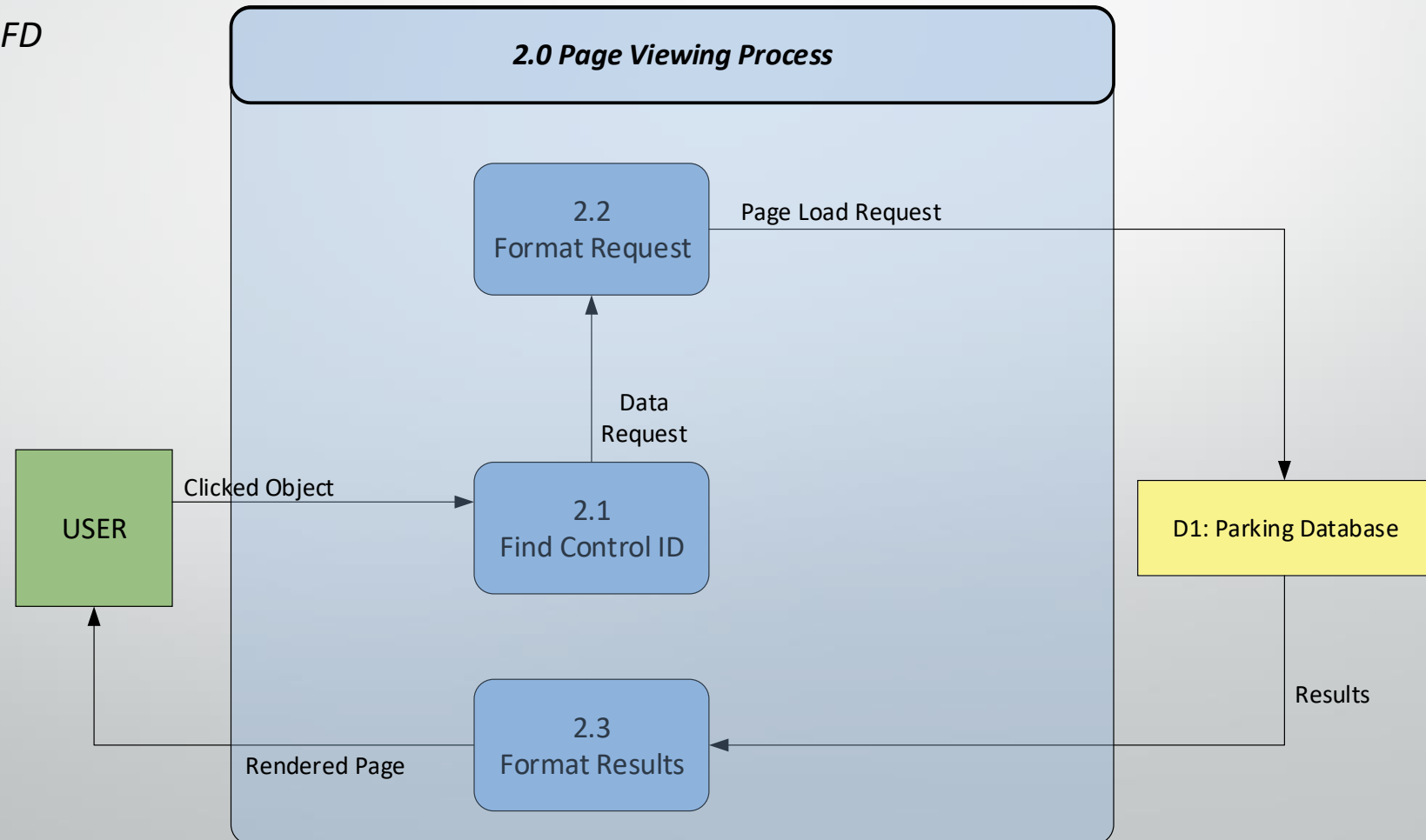
Level 2 DFD



Analysis: PROCESS MODELING CONT'D

Parking Services System

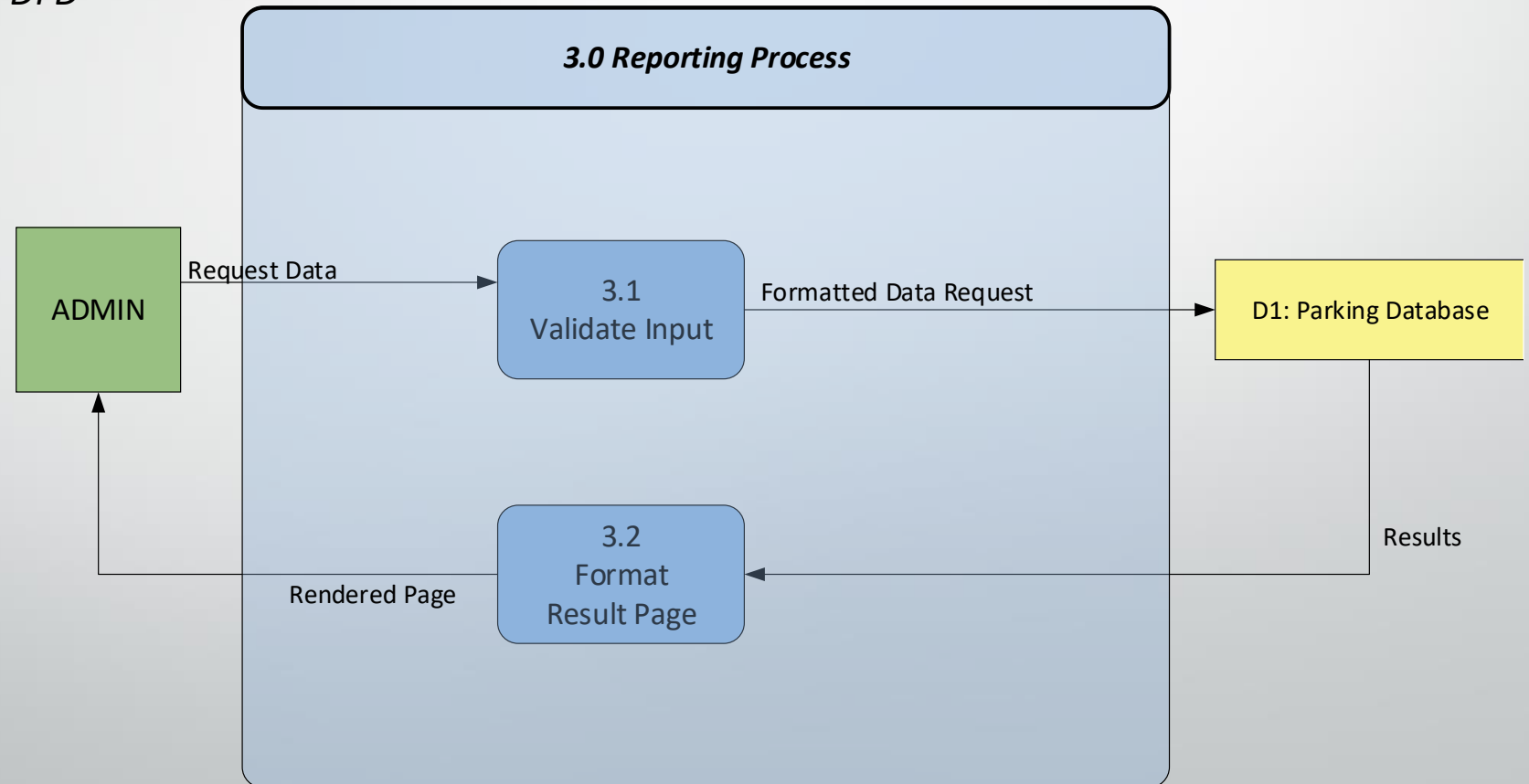
Level 2 DFD



Analysis: PROCESS MODELING CONT'D

Parking Services System

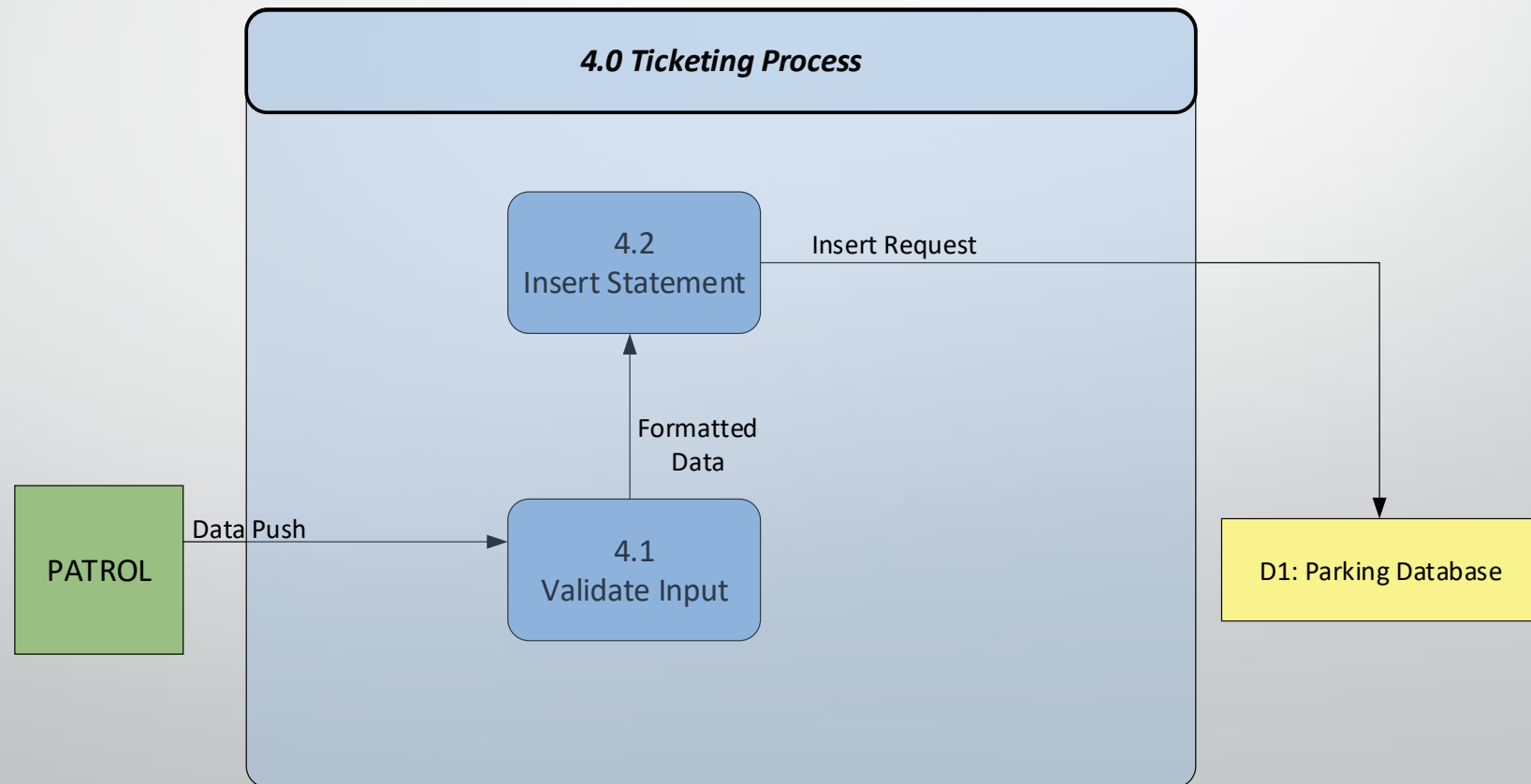
Level 2 DFD



Analysis: PROCESS MODELING CONT'D

Parking Services System

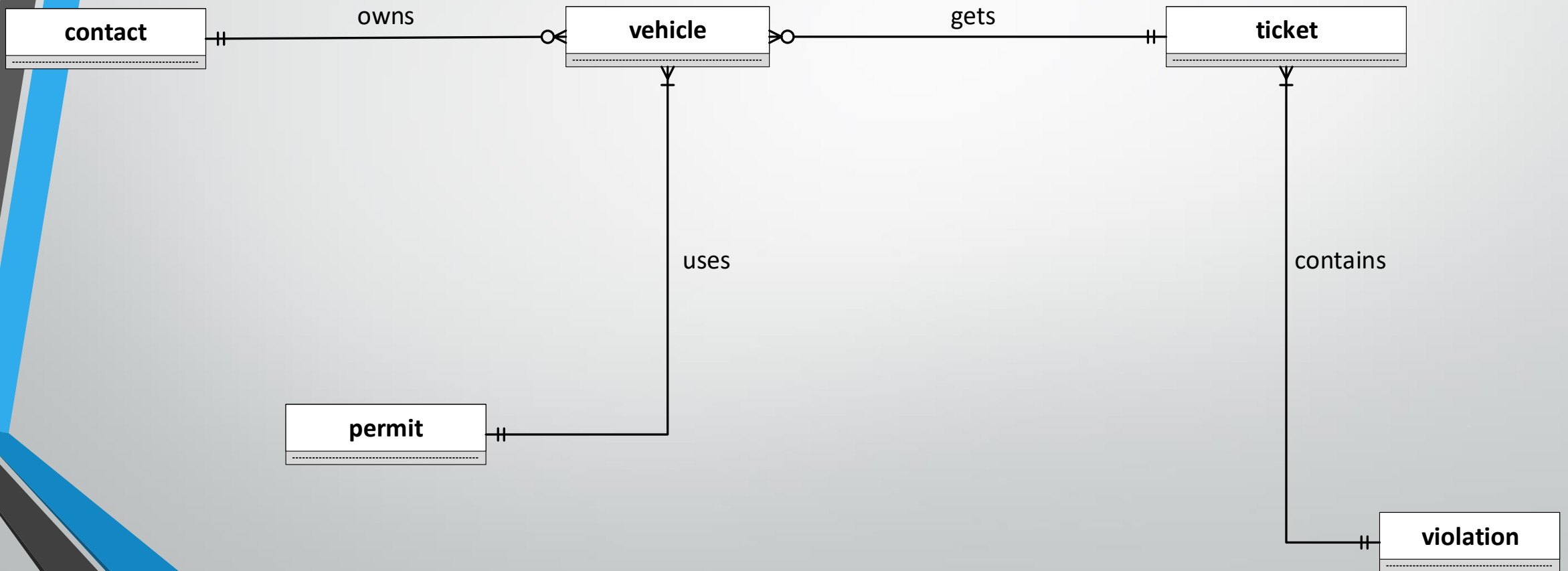
Level 2 DFD



Analysis: *Structuring Requirements* – CONCEPTUAL DATA MODELING

Parking Services System

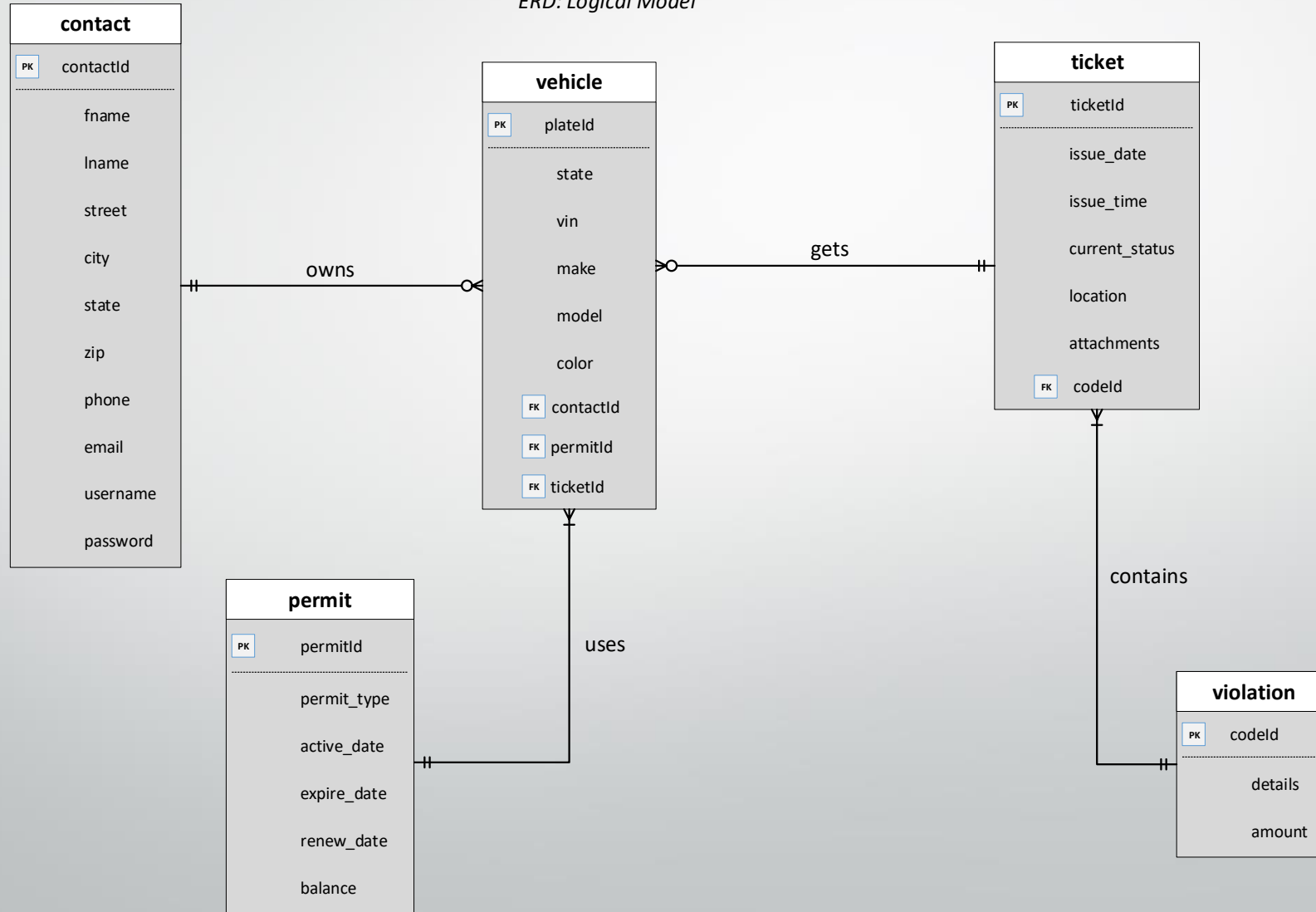
ERD: Conceptual Model



Analysis: CONCEPTUAL MODELING CONT'D

Parking Services System

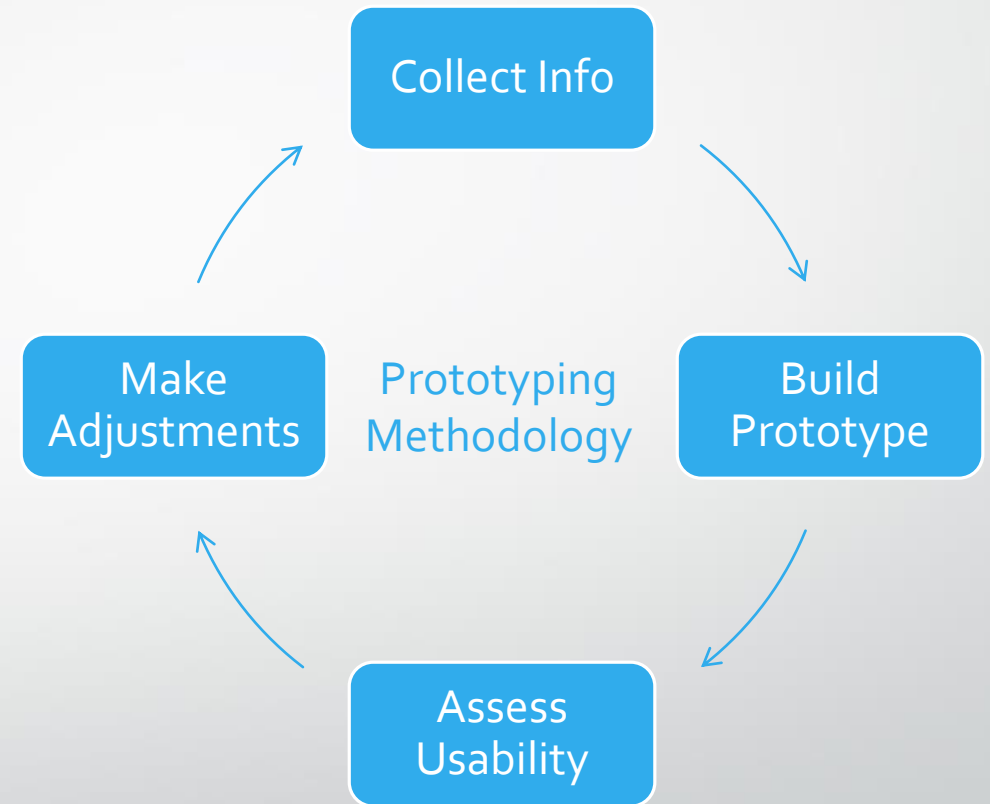
ERD: Logical Model



Design: *Front-End*

Designing Forms & Reports

Who will use?	Purpose?	When are they used?
How many use?	Style Conformity?	Where to deliver?



Design: *Front-End Cont'd*

Requirements		
Secure Access	Check Balances	Permit Request

Login

Label prompts

Textbox for username
& password input

Submit Button

Check Balances

Initiation Button

Labels/Tables
for Data Display

Permit Request

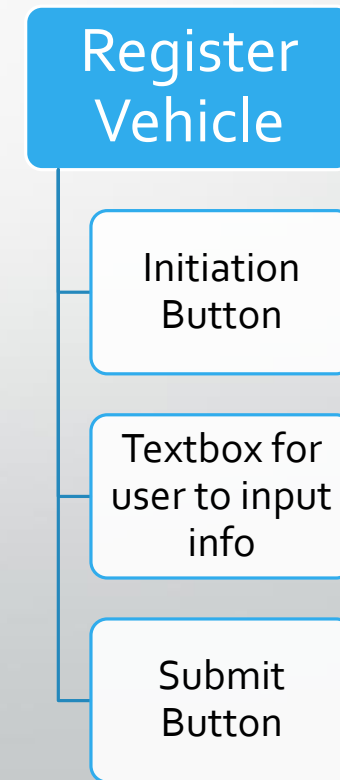
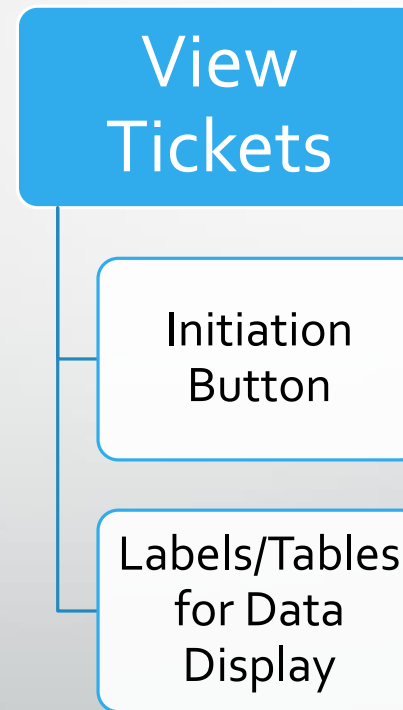
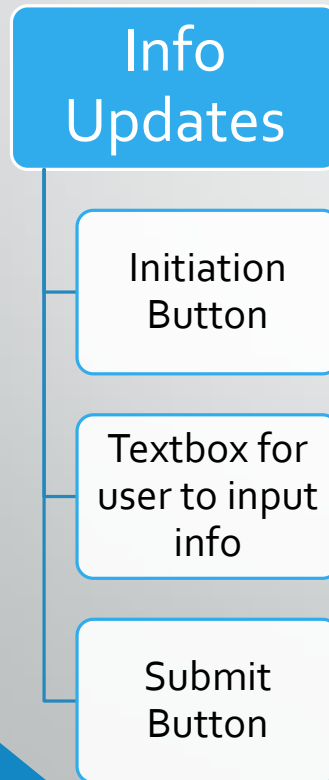
Initiation Button

Selectable list of
users registered
vehicles

Submit Button

Design: *Front-End Cont'd*

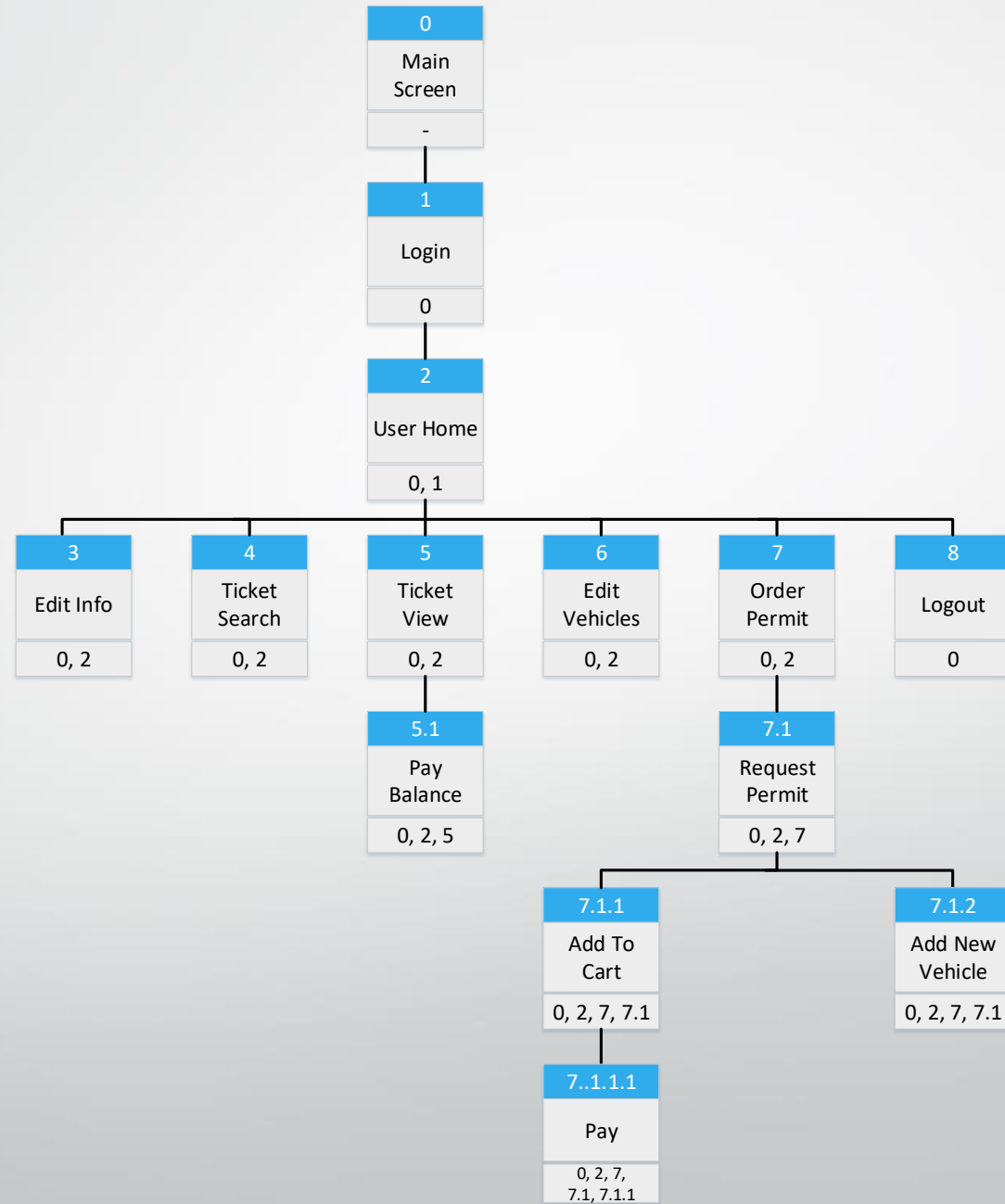
Requirements		
Info Updates	View Citations	Register Vehicles



Design:

Front-End
Cont'd

Dialogue Diagram



Design: Server Side

CONTACT

Attribute Name	Data Type	Field Length	Key	Description	Format	Required?	Constraints	Example
contactId	CHAR	9	PK	Uniquely identifies a contact	Not Null	Y	-	900654321
fname	VARCHAR	50	-	Describes the first name of a contact	Not Null	Y	-	John
lname	VARCHAR	50	-	Describes the last name of a contact	Not Null	Y	-	Doe
street	VARCHAR	50	-	Describes the street address for a contact	Not Null	Y	-	1234 Park Street Lane
city	VARCHAR	50	-	Describes the city for a contact	Not Null	Y	-	Cityville
states	CHAR	2	-	Describes the state for a contact	Not Null	Y	-	GA
zip	CHAR	5	-	Describes the postal code for a contact	Not Null	Y	-	30533
phone	VARCHAR	12	-	Describes the phone number for a contact	Not Null	Y	-	123-456-7890
email	VARCHAR	50	-	Describes the email address for a contact	Not Null	Y	-	JDOE@ung.edu
uname	VARCHAR	25	-	Describes the contacts login username	Not Null	Y	-	user1234
pword	VARCHAR	25	-	Describes the contacts login password	Not Null	Y	-	password123

TICKET

Attribute Name	Data Type	Field Length	Key	Description	Format	Required?	Constraints	Example
ticketId	INT	<i>identity</i>	PK	Uniquely identifies a ticket	Not Null	Y	-	18Y01234
issue_date	DATE	-	-	Describes the date a ticket was issued	Not Null	Y	-	01-02-2018
issue_time	TIME	-	-	Describes the time a ticket was issued	Not Null	Y	-	11:12 AM
current_status	VARCHAR	6	-	Describes the ticket as being paid or unpaid	Not Null	Y	-	Paid
location	VARCHAR	25	-	Describes the lot & campus for an issued ticket	Not Null	Y	-	L16 CHURCH ST. LOT
attachments	VARCHAR	255	-	Displays an image of the vehicle	Not Null	Y	-	0xFFD8FFE00104A460. . .
codId	VARCHAR	6	FK	Uniquely identifies a violation	Not Null	Y	violation.codId	32

VEHICLE

Attribute Name	Data Type	Field Length	Key	Description	Format	Required?	Constraints	Example
plateId	CHAR	7	PK	Uniquely identifies a vehicle	Not Null	Y	-	ABC1234
state	CHAR	2	-	Describes the state for a license plate	Not Null	Y	-	GA
vin	VARCHAR	17	-	Describes the vehicle identification number	Nullable	Y	-	1HGBH41JXMN109186
make	VARCHAR	25	-	Describes the vehicle manufacturer name	Not Null	Y	-	HONDA
model	VARCHAR	25	-	Describes the vehicle model name	Not Null	Y	-	CIVIC
color	VARCHAR	25	-	Describes the vehicle color	Not Null	Y	-	BLACK
contactId	CHAR	9	FK1	Uniquely identifies a contact	Not Null	Y	contact.contactId	900654321
ticketId	INT	<i>identity</i>	FK2	Uniquely identifies a ticket	Not Null	Y	ticket.ticketId	7654
permitId	INT	<i>identity</i>	FK3	Uniquely identifies a permit number	Not Null	Y	permit.permitId	325

PERMIT

Attribute Name	Data Type	Field Length	Key	Description	Format	Required?	Constraints	Example
permitId	INT	<i>identity</i>	PK	Uniquely identifies a permit number	Not Null	Y	-	S1801234
permit_type	VARCHAR	25	-	Describes the type of permit	Not Null	Y	-	Student-Dahl. Commuter
active_date	DATE	-	-	Describes the permit activation date	NOT NULL	Y	-	01-02-2018
expire_date	DATE	-	-	Describes the permit expiration date	Not Null	Y	-	01-02-2018
renew_date	DATE	-	-	Describes the recommended renew date	Nullable	N	-	01-02-2018
balance	SMALLMONEY	-	-	Describes the permit user balance	Not Null	Y	-	\$0.00

VIOLATION

Attribute Name	Data Type	Field Length	Key	Description	Format	Required?	Constraints	Example
codId	VARCHAR	6	PK	Uniquely identifies a violation	Not Null	Y	-	32
detail	VARCHAR	50	-	Describes the rule broken causing violation	Not Null	Y	-	EXPIRED PERMIT
amount	SMALLMONEY	-	-	Describes the money owed for a violation	Not Null	Y	-	\$15.00

Design: *Server Side Cont'd*

```
--CONTACT
CREATE TABLE contact
(
    contactId INT IDENTITY(1,1) NOT NULL,
    fname VARCHAR(50),
    lname VARCHAR(50),
    street VARCHAR(50),
    city VARCHAR(50),
    states CHARACTER(2),
    zip CHARACTER(5),
    phone VARCHAR(12),
    email VARCHAR(50),
    uname varchar(25),
    pword varchar(25),
    plateId varchar(20),
    PRIMARY KEY(contactId)
);
```

```
--GET USER
GO
CREATE PROCEDURE get_contact
    @uname NVARCHAR(20),
    @pword NVARCHAR(20)
AS
BEGIN
    SELECT contactId
    FROM contact
    WHERE uname = @uname AND pword = @pword
END;
```

```
--Add Foreign Key Constraints
--VEHICLE
ALTER TABLE vehicle
    ADD CONSTRAINT fk_vehiclecontactid
    FOREIGN KEY (contactId) REFERENCES contact(contactId);

--
ALTER TABLE vehicle
    ADD CONSTRAINT fk_vehiclepermitid
    FOREIGN KEY (permitId) REFERENCES permit(permitId);

--
ALTER TABLE vehicle
    ADD CONSTRAINT fk_vehicleticketid
    FOREIGN KEY (ticketId) REFERENCES ticket(ticketId);
```

```
--GET PERMIT
GO
CREATE PROCEDURE get_permit
    @contactId INT
AS
BEGIN
    SELECT permit.permitId AS "Permit #", permit.permit_type AS "Permit Type",
        CONVERT(NVARCHAR,permit.expire_date,101) AS "Expire Date",
        CONVERT(NVARCHAR,permit.renew_date,101) AS "Renew Date",
        permit.balance AS 'Balance'
    FROM permit
        INNER JOIN vehicle ON vehicle.permitId = permit.permitId
        INNER JOIN contact ON contact.contactId = vehicle.contactId
    WHERE vehicle.contactId = @contactId
END;
```

Implementation & Operation

<http://ungparkingsystem.azurewebsites.net>

No longer active

- Install Hardware & Software
 - Programming
 - User Training
 - Documentation



Questions?