```
from flask_wtf import FlaskForm
from wtforms import (StringField, PasswordField, TextAreaField,
                     IntegerField, SelectField, RadioField, validators,
                         SubmitField)
from wtforms.validators import (DataRequired, Regexp,
                                ValidationError, Email, Length, EqualTo)

from models import User


def check_isbn(form, field):
    if len(str(abs(field.data))) != 13:
        raise ValidationError("Please enter a 13 digit code for ISBN 13")


def name_does_not_exist(form, field):
    if not (User.select().where(User.username == field.data).exists()):
        raise ValidationError("This user does not exist")


def name_exists(form, field):
    if User.select().where(User.username == field.data).exists():
        raise ValidationError("Sorry, this username is already taken. Try
            again.")


def email_exists(form, field):
    if User.select().where(User.email == field.data).exists():
        raise ValidationError("Sorry, this email is already taken. Try again.")


class LoginForm(FlaskForm):
    email = StringField(
        'Email',
        validators=[
            DataRequired(),
            Email()
        ]
    )

    password = PasswordField(
        'Password',
        validators=[
            DataRequired()
        ]
    )


class RegistrationForm(FlaskForm):
    username = StringField(
        'Username',
        validators=[
            DataRequired(),
            Regexp(
                r'^[a-zA-Z0-9_]+$',
                message="Username can only be one word.  Letters, numbers and
```

```
                    underscores only!"
            ),
            name_exists
        ]
    )

    email = StringField(
        'Email',
        validators=[
            DataRequired(),
            Email(),
            email_exists
        ]
    )

    password = PasswordField(
        'Password',
        validators=[
            DataRequired(),
            Length(min=5, max=100),
            EqualTo('confirm_password', message='Passwords must match!')
        ]
    )

    confirm_password = PasswordField(
        'Confirm Password',
        validators=[
            DataRequired()
        ]
    )


class UpdateUsername(FlaskForm):
    username = StringField(
        'New Username',
        validators=[
            DataRequired(),
            Regexp(
                r'^[a-zA-Z0-9_]+$',
                message="Username can only be one word.  Letters, numbers and
                    underscores only!"
            ),
            name_exists
        ]
    )


class UpdateEmail(FlaskForm):
    email = StringField(
        'New Email',
        validators=[
            DataRequired(),
            Email(),
            email_exists
        ]
    )
```

```
class UpdatePassword(FlaskForm):
    password = PasswordField(
        'New Password',
        validators=[
            DataRequired(),
            Length(min=5, max=100),
            EqualTo('confirm_password', message='Passwords must match!')
        ]
    )

    confirm_password = PasswordField(
        'Confirm New Password',
        validators=[
            DataRequired()
        ]
    )


class MessageForm(FlaskForm):
    to_user = StringField(
        'Username',
        validators=[
            DataRequired(),
            Regexp(
                r'^[a-zA-Z0-9_]+$',
                message="Username can only be one word.  Letters, numbers and
                    underscores only!"
            ),
            name_does_not_exist
        ]
    )
    content = TextAreaField(
        'Message Area!',
        validators=[
            DataRequired()
        ]
    )


class BookPostForm(FlaskForm):
    title = StringField(
        'Title',
        validators=[
            DataRequired(),
        ]
    )

    isbn = IntegerField(
        'ISBN 13',
        validators=[
            DataRequired(),
            check_isbn
        ]
    )
```

```python
    course = StringField(
        'Course'
    )

    major = StringField(
        'Major'
    )
    condition = SelectField(
        'Book Condition',
        validators=[DataRequired()],
        choices=[('new', 'New'), ('good', 'Good'), ('fair', 'Fair'), ('poor',
            'Poor')]
    )
    comment = TextAreaField(
        'Comments',
        validators=[
            Length(max=150)
        ]
    )


class SearchForm(FlaskForm):
    category = SelectField(
        'Book Condition',
        validators=[DataRequired()],
        choices=[('title', 'Title'), ('isbn', 'ISBN 13'), ('course', 'Course'),
            ('major', 'Major'), ('user', 'User')]
    )
    search_field = StringField(
        'Search',
        validators=[DataRequired()]
    )


class SurveyForm(FlaskForm):
    # User to be rated
    user = StringField(
        'User',
        validators=[DataRequired(), name_does_not_exist]
    )
    rating = RadioField(
        'Rating',
        validators=[DataRequired()],
        choices=[('Thumbs up', 'approve'), ('Thumbs down', 'disapprove')]
    )
    comment = StringField(
        'Comment',
        validators=[
            Length(max=250)
        ]
    )


class ConversationForm(FlaskForm):
    user = StringField(
        'User',
        validators=[
            DataRequired(),
            name_exists
        ]
    )
```